CS589: Machine Learning - Fall 2019

Homework 5: Unsupervised learning

Assigned: 23^{rd} April, 2019; Due: 2^{nd} May, 2019

Getting Started: Please install the sklearn package for Python 3.6. Unzipping this folder will create the directory structure shown below. You will submit your code under the Submission/Code directory.

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

In this assignment, you will implement several unsupervised learning algorithms and you will use them to compress images.

Deliverables: This assignment has two types of deliverables: a report and code files.

- **Report:** The solution report will give your answers to the homework questions (listed below). Try to keep the maximum length of the report to 5 pages in 11 point font, including all figures and tables. You can use any software to create your report, but your report must be submitted in PDF format. Please ensure that the answers in the report follow the same order as given in this document.
- **Code:** The second deliverable is the code that you wrote to answer the questions, which will involve implementing a sampling algorithm. Your code must be Python 2.7 (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 re-produce all of your experimental results exactly by running *python run_me.py* file from the Submissions/Code directory. 10% of your assignment grade is based on code quality.

Submitting Solutions: 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. If you used Python to generate report figures, place them in Submission/Figures. 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-Unsupervised-Learning' 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-Unsupervised-Learning-PDF' 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, web pages, etc.) or other students is considered cheating. Sharing your solutions with other students is considered cheating. Posting your code to public repositories like GitHub 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: In contrary of supervised learning (classification, regression), unsupervised learning algorithms attempt to learn some structure of the data using unlabeled samples. There are several algorithms within the unsupervised learning scope: principal component analysis (PCA), k-means, independent component analysis (ICA), and density estimation, among others. In this project you will use K-Means to compress images.

Algorithm and data:

k-means: clustering algorithm; it finds centroids and assign each sample to one and only one of these according to some criteria. You will use k-means to compress the following image

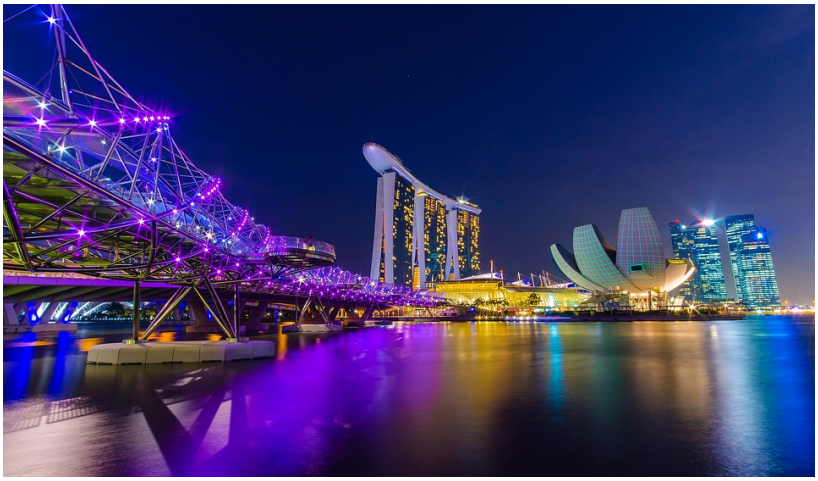

Figure 1: Image to compress using k-means

Questions:

1. (*100 points*) K-means:

(10) 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.

(15) b. Another issue with k-means is that the random initialization of the centroids can sometimes lead to "poor" clusters. A possible solution to this problem is presented in the algorithm called k-means++. Briefly explain the idea behind this algorithm.

(40) c. You are given an RGB image *test_image.jpg*. Each pixel can be seen as a sample of dimension 3 (3 integers between 0 and 255, one for each component RGB). Take each pixel as a sample, and apply k-means using k centroids, for $k = \{2, 5, 10, 25, 50, 100, 200\}$ (note that in this case each centroid represents an RGB color. Thus, k is the number of colors in the compressed images). Replace each pixel in the original image for the centroid assigned to it. Show the original image in the report and the reconstructed images for each value of k. An example of a reconstructed image using 15 clusters is shown in 2.

k	Compression rate
2	
5	
10	
25	
50	
100	
200	

(35) d. For each value of k show, using a table as the one shown above, the recontruction error for each value of k. Also, using another table, show the compression rate for each k. 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).

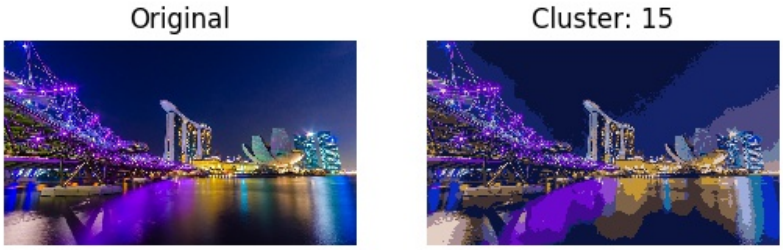

Figure 2: Example of reconstructed image using 15 clusters