CS589: Machine Learning — Spring 2018

Extra Credit: Clustering

Assigned: April 26^{th} – Due: May 7^{th}

Getting Started: In this assignment, you will perform clustering to compress images. **Please install Python 3.6 on your personal machine and please use only scikit-learn libraries for this assignment**. Download the homework file EC.zip via Moodle. Unzipping this folder will create the directory structure shown below,

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

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 deliverables:

- **Report:** The solution report will give your answers to the homework questions (listed below). Please keep the maximum length of the report to four pages in 11 point font, including all figures and tables. Reports longer than four pages will only be graded up until the first for pages. You can use any software to create the report (as opposed to your analysis), 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 clustering. 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 Gradescope. Place your final code in Submission/Code. If you generated any figures place them under 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 'Extra Credit — 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 'Extra Credit — 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 such as GitHub and Stack Overflow is also considered cheating. Any detected cheating will result in a grade of 0 on the assignment for all students involved, and potentially a grade of F in the course.

Task: Contrary to supervised learning (e.g., classification and regression), unsupervised learning algorithms learn patterns from unlabeled examples. For this project, you will use hierarchical clustering and k-means clustering to compress images. We provide an RGB image *umass_campus.jpg* of the UMass campus as a $400 \times 400 \times 3$ matrix. Each pixel can be seen as a sample of dimension three (three integers between 0 and 255, one for each component of RGB). For this question you will treat each pixel as a data instance.

Algorithms and data:

Hierarchical agglomerative clustering (HAC) constructs a complete hierarchy over a set of data instances such that a large number of different clusters can be constructed by selecting different 'levels' of the hierarchy. You will use hierarchical clustering to compress an image of the UMass campus as shown in Figure 1.

K-means clustering finds centroids of clusters and assigns each sample to one of these clusters based on a distance function. You will also use k-means to compress an image of the UMass campus as shown in Figure 1.

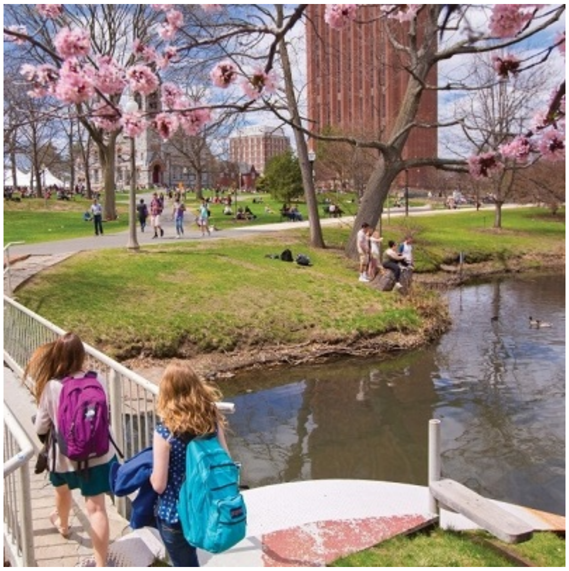

Figure 1: Image to compress using k-means

Questions:

1. *Perform k-means clustering* (30%) — Apply k-means clustering where k is in the range {2, 5, 10, 25, 50, 75, 100, 200}. Replace each pixel in the original image with the centroid of the cluster to which that pixel is assigned. Create a 3×3 grid plot of the original image along with eight reconstructed

images corresponding to eight different values of k. Be certain 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.

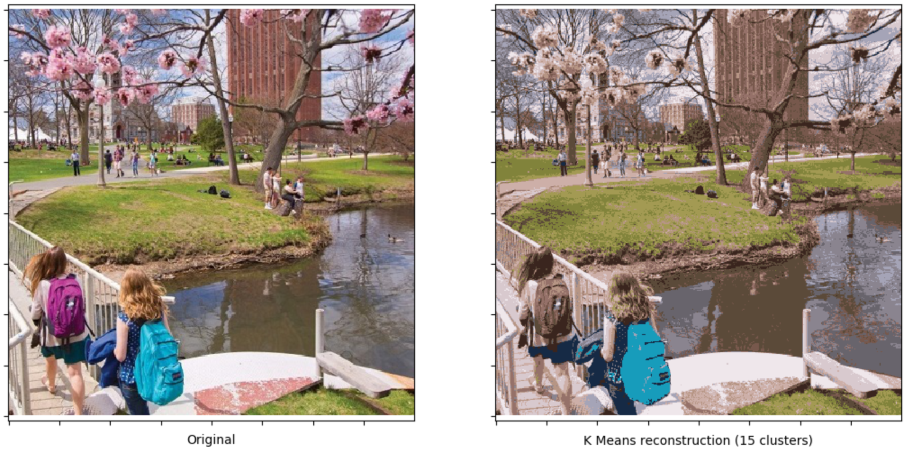

Figure 2: Example of reconstructed image using 15 clusters

2. *Select functions for HAC* (10%) — Select both the distance function and the linkage function (what scikit-learn calls the 'affinity' and 'linkage' parameters, respectively) necessary to apply HAC. Briefly describe a justification for selecting these particular functions (e.g., "tried several options and this one had the best compression" or "Only this function gave substantially different results than k-means clustering").

3. *Perform HAC* (30%) — Apply HAC and obtain k clusters in the range $\{2, 5, 10, 25, 50, 75, 100, 200\}$ (note that scikit-learn's 'AgglomerativeClustering' has a parameter 'n_cluster' for this purpose). Replace each pixel in the original image with the centroid RGB value of the points in the cluster of that pixel. As before, create a 3×3 grid plot of the original image along with eight reconstructed images corresponding to eight different values of k. Be certain to label the images.

4. *Create a table reporting results* (20%) — Create a table similar to the one below that reports the reconstruction error for each clustering method for each value of k. The reconstruction error is $\sqrt{\text{mean}(\mathbf{X} - \mathbf{X_{recon}})^2}$, for the **entire dataset**. For each clustering method and value of k, report the reconstruction error using a table similar to the one below.

k	K-means Reconstruction Error	HAC Reconstruction Error
2		
5		
10		
25		
50		
75		
100		
200		

Table 1: Reconstruction error for clustering applied to the UMass Photo

5. *Select and explain an overall method* (10%) — Based on the images, values of reconstruction error, and the elbow method (see below), select and briefly describe in a few sentences your recommendation for the algorithm and value of k that the UMass website should use for image compression when delivering images to mobile phones. The elbow method is a very simple approach to selecting a value of k (see the Wikipedia entry on 'Elbow method (clustering)' for details).