CS589: Machine Learning - Spring 2020

Homework 5: Unsupervised learning

Assigned: 22^{rd} April, 2020; Due: 1^{st} May, 2020

In this assignment, you will implement two very simple versions of lossy image compression– one based on PCA and one based on K-means.
Getting Started: Please install the sklearn package for Python 3.7. 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
```

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, including all figures and tables. Reports longer than 5 pages will only be graded up until the first 5 pages. 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 implementing a sampling algorithm. Your code must be Python 3.7 (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 `run_me.py` file from the `Submissions/Code` directory. 10% of your assignment grade is based on code quality.

Submitting Deliverables: When you complete the assignment, you will upload your report and your code using the `Gradescope.com` service. Here are the steps:

1. Place your final code in `Submission/Code`.
2. Create a **zip** file of your submission directory, `Submission.zip` (No rar, tar or other formats).
3. Upload this single zip file on Gradescope as your solution to the `HW05-Unsupervised-Programming` assignment.
4. Upload your pdf report to the `HW05-Unsupervised-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.

5. The submission time for your assignment is considered to be the latest of the submission timestamps of your code and report.

Academic Honesty Policy: You are required to list the names of anyone you discuss problems with on the first page of your solutions. This includes teaching assistants or instructors. Copying any solution materials from external sources (books, web pages, etc.) or other students is considered cheating. To emphasize: no detectable copying is acceptable, even, e.g., copying a single sentence from an outside source. Sharing your code or solutions with other students is also considered cheating. Any detected cheating will result in a grade of -100% on the assignment for all students involved (negative credit), and potentially a grade of F in the course.

Task:

1. (*0 points*) **Collaboration statement**:
Please list the names of anyone you discussed the assignment with, including TAs or instructors.

2. (*55 points*) **PCA:** For this problem you are given 100 images, each containing a face like the following:

Figure 1: Sample of the faces dataset

In this problem you will use PCA to compress this set of images.

(10) a. Suppose that you are given a dataset with N samples, each of dimension D. Show that the direction that maximizes the variance (minimizes reconstruction error) of the data is generated by the eigenvector associated to the largest eigenvalue of the estimated covariance matrix of the data.

Hint: There are many ways of doing this. One is to solve

$$\min_{w} \frac{1}{N} \sum_{n=1}^{N} ||x^{(n)} - \hat{x}^{(n)}||^2 \text{ subject to } ||w|| = 1$$

with $x^{(n)}$ being the n-th sample placed as a column vector, and $\hat{x}^{(n)} = (w \cdot x^{(n)})w$ being the projection of the n-th sample onto the direction w. Recall that the estimated covariance matrix of the data is $C = \frac{1}{N} \sum_{n=1}^{N} x^{(n)} \, x^{(n)^T}$.

(5) b. Show that the subspace of dimension two that maximizes the variance (minimizes the reconstruction error) of the data is generated by the two eigenvectors associated to the largest two eigenvalues of the estimated covariance matrix of the data.

(8) c. Let x be a sample and let x_i represent the i-th component of x. Suppose that for every sample in the dataset, there exist a set of constants $a_1, a_2, \cdots, a_{D-1}$ such that the last component of x is

$$x_D = \sum_{i=1}^{D-1} a_i \, x_i.$$

What is the minimum number of directions (eigenvectors of the estimated covariance matrix) needed to store the data perfectly (ie. no loss of information)? Explain in at most 4 sentences.

Hint: Thinking about the specific case when $D = 3$ might help.

(22) d. You are given a set of 100 images (50×50 pixels, gray scale) containing faces. Each sample a 50x50 image which can be stored as a vector $x^{(n)} \in \mathbb{R}^{2500}$. Using these the covariance matrix of the data is $C = \frac{1}{N} \sum_{n=1}^{N} x^{(n)} \, x^{(n)^T} \in \mathbb{R}^{2500 \times 2500}$. For each value of $k \in \{3, 5, 10, 30, 50, 100, 150, 300\}$, find the k eigenvectors associated with the k largest eigenvalues of X. Call these vectors $w_1, \cdots, w_k$. Project the data onto them. Thus, the compressed representation of $x^{(n)}$ is the set of numbers $y_1^{(n)}, \cdots, y_k^{(n)}$ where

$$y_j^{(n)} = w_j \cdot x^{(n)}.$$

You can then approximately reconstruct $x^{(n)}$ from the compressed representation as

$$\hat{x}^{(n)} = \sum_{j=1}^{k} w_j \cdot y_j^{(n)}.$$

Show in the report the original `face.png`, and for each k show the image obtained after projecting it. Plot the average squared reconstruction error for the whole dataset as a function of k. For this question you have to implement the algorithm using only a package that computes the eigenvalues-eigenvectors decomposition of a matrix (ie. NO package that implements PCA).

(10) e. For each value of k show, using a table, the compression rate achieved when projecting the whole dataset into the new subspace. Use the following table as a template.

k	Compression rate
3	
5	
10	
30	
50	
100	
150	
300	

The compression rate is the memory required to store the compressed images divided by the memory required to store the original images. In this case the former includes the space required to store the k

principal components (vectors, used for all images), and the k factors (the $\alpha_{i,j}$) for each sample, and the latter is the sum of memory required for each image.

3. (*45 points*) **K-means:** For this problem, you will use k-means to compress the following (single) image (available in the `Data` folder:

Figure 2: Image to compress using 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. 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.

(20) c. You are given an RGB image `shopping-street.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 Fig. 3.

(b): Original

(d): Reconstruction

Figure 3: Example of reconstructed image using 15 clusters

(15) d. For each value of k show, using a table as the one shown above, the reconstruction 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 $\lceil \log_2 k \rceil$ bits (for each pixel in the image you store the index of the centroid assigned).