CSCI 3320: Fundamental of Machine Learning **2018-2019 Term 2**

Programming Assignment 3

Instructor: Prof. John C.S. Lui Due: 23:59 on Wed. Apr. 24, 2019

1 Introduction

k-means clustering is a method of vector quantization. It aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. When handling the high-dimensional input data, to make the clustering more efficient, we can apply Principal Component Analysis (PCA) to perform the dimension reduction for the input data. PCA provides a way to create a low dimensional representation of data. This can not only reduce the complexity of the clustering, but also improve the clustering result by the noise reduction. This programming assignment is provided to help you gain a better understanding of these two methods and by applying them to an image data application. More specifically, this programming assignment consists of the following two parts.

- The first part will guide you through a complete version of **Principal Component Analysis (PCA)**. First, you will learn to implement your own PCA, and compare with the scikit-learn programming interface for the PCA task. Then, you will have an idea of solving practical problems with PCA by performing it on the image data.
- The second part is an exercise about **k-means clustering**. The purpose is to help you understand the process of the k-means clustering and different performance evaluation metrics. You will also learn the scikit-learn programming interface for k-means clustering.

1.1 File Descriptions

To start, you need to download the `asgn3.zip` file from the course website. In `asgn3.zip`, we provide the following files for you:

- `ex1.py`: contains some python scripts for you to learn to implement your own PCA.
- `ex2.py`: contains some python scripts for you to learn to apply PCA on the image data.
- `mnist-subset.zip`: contains images of handwritten digits 0, 1, 2, which is a subset of the LeCun's MNIST dataset.
- `ex3.py`: contains some python scripts for you to learn to perform k-means clustering on the synthetic dataset.

- `ex4.py`: contains some python scripts for you to learn to perform k-means clustering on the image data.
- `t10k-images.idx3-ubyte` and `t10k-images.idx3-ubyte`: contains images of handwritten digits, which is a subset of the LeCun's MNIST dataset.

Note: Please do **not** change filenames (`*.py`) of files described above.

2 Principal Component Analysis (PCA)[1]

2.1 Implement PCA(20%)

Our first problem is to perform PCA on the created dataset. The purpose is to help you understand details of PCA. Please refer to chapter 6 (Dimension Reduction) of the lecture notes. You need to finish the following tasks:

1. **Write Your Own PCA**. (10%)
   Read lecture notes carefully, and implement your own PCA. You need to complete the function `pca(X)` in the `ex1.py` script. The input is the matrix $\boldsymbol{X}$ in which each row represents a sample. The output should be `[V, D]`, where `V` is a matrix containing all eigenvectors (whose eigenvalues are in a decreasing order) and `D` should be a column vector containing all eigenvalues (in a descent order). You can compare your result with the one of PCA in the scikit-learn[2].

2. **Plot Eigenvalues and POV**. (10%)
   In `ex1.py`, we generate a dataset $\boldsymbol{X}$, which is a 1000×9 matrix. Perform PCA on this matrix and plot all eigenvalues in a decreasing order. Assume the descent ordered eigenvalues are $\lambda_1, \lambda_2, \cdots, \lambda_N$. The x-axis should be $1, 2, \cdots, N$ (the order of eigenvalues) and y-axis should be the eigenvalues (refer to pages 20 in chapter 6). Use the definition of Proportion of Variance (POV) explained in the lecture notes, plot POV v.s. the order of the eigenvalues (refer to pages 20 in Lecture 8 (Chapter 6)).

Note: You need to submit `ex1.py` in task 2.1. Your `ex1.py` file should output the result of `[V, D]` to the prompt and plot the figure of eigenvalues v.s. the order of the eigenvalues and the figure of POV v.s. the order of the eigenvalues. Make sure your python script has no syntax error before submission.

2.2 Perform PCA on Image Data(25%)

In this part, you are required to perform PCA on images of handwritten digits. In `asgn3.zip`, we provide `mnist-subset.zip` which is a subset of the LeCun's MNIST dataset containing

[1] More details of this part you can refer to the tutorial slides `tutorial 7.ppt`.

[2] `http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html`

images of handwritten digits 0, 1, 2. We already provide the code to read images into a matrix $\boldsymbol{X}$ to represent them in python script `ex2.py`. Each row of the matrix $\boldsymbol{X}$ represents an image. You are required to finish the following task.

1. **Plot the Eigen Images**. (10%)
   Perform PCA on the matrix $\boldsymbol{X}$. You can use PCA implemented by yourself or the one provided in scikit-learn. Plot **eigenvectors** corresponding to **the largest 10 eigenvalues**. Combine and save figures in a single image file `eigenimages.jpg`. We also provide the sample code for presenting **vectors** as an image. For PCA on images data, you can read the example in scikit-learn[3].

2. **Plot POV**. (15%)
   Plot POV v.s. the order of eigenvalues as in Task A. Combine and save figures in file `digit_pov.jpg`. You also need to answer that to achieve POV larger than 0.95, how much dimensions should we preserve? Create a file `description2.txt`(`or description2.pdf`) to describe your answers and your understanding of the result.

Note: You need to submit the python script `ex2.py` which includes the implementation of task 2.2.. Besides, you also need to submit `eigenimages.jpg, digit_pov.jpg` and `description2.txt` (`or description2.pdf`) in this task.

3 k-Means Clustering

3.1 k-Means Clustering on the Synthetic Dataset(25%)

Our problem in this task is to perform k-means clustering on the synthetic dataset. The purpose is to help you understand the process of k-means clustering and different performance evaluation metrics. You need to finish the following tasks. Please use `ex3.py` as your starting point.

1. **Creat dataset**. (5%)
   Please read how we generate the datasets in `creat_dataset()`, and see what are centers of clusters. Complete `creat_datasets()` and plot data points in a scatter plot. Use different colors to represent clusters.

2. **k-means clustering**. (10%)
   Complete the function `my_clustering(X, y, n_clusters)`.

   - **Inputs.** Sample points $(\boldsymbol{X}, y)$ and the number of clustering.
   - **Tasks.** Cluster samples into `n_clusters` clusters. Print out all cluster centers. Plot all clusters formed, and use different colors to represent clusters defined by k-means. Draw a marker (e.g., a circle or the cluster id) at each cluster center.

[3] `http://scikit-learn.org/stable/auto_examples/applications/face_recognition.html`

- **Outputs.** Return the following four evaluation metrics: *Adjusted Rand index*, *Mutual Information based scores*, *V-measure*, and *Silhouette Coefficient*. For details about these metrics, you can read the documentation of scikit-learn [4].

3. **Select the number of clusters**. (10%)
   Plot scores of all four evaluation metrics as functions of `n_clusters`. What's the best number of clusters? Answer this in `description3.txt` (`or description3.pdf`).

Note: You need to submit `ex3.py` and `description3.txt` (`or description3.pdf`) for task 3.1. Your `ex3.py` script should output cluster centers and values of four evaluation metrics mentioned above. Also, it should plot formed clusters and scores of all four evaluation metrics as functions of `n_clusters` as mentioned above.

3.2 k-Means Clustering on the Image Dataset(30%)

In this part, we want to cluster images of handwritten digits. In `asgn3.zip`, we provide two input files `t10k-images.idx3-ubyte` and `t10k-labels.idx1-ubyte`. They are subsets of the LeCun's MNIST dataset containing images of handwritten digits. You are required to finish the following tasks. Please use `ex4.py` as your starting point.

1. **Load the dataset**. (5%)
   We provide codes to load images and their labels into a matrix $\boldsymbol{X}$ and a label vector $\boldsymbol{y}$ to represent them in python script `ex4.py`. Each row of the matrix $\boldsymbol{X}$ represents an image. What are the number of samples and the number of features? Answer them in `description4.txt` (`or description4.pdf`).

2. **PCA**. (5%)
   Perform PCA on the matrix $\boldsymbol{X}$ (requirement: $POV \geq 0.95$).

3. **k-means clustering**. (10%)
   Complete `my_clustering(X, y, n_clusters)`.

   - **Inputs.** Sample points $(\boldsymbol{X}, y)$ and the number of clustering `n_clusters`.
   - **Tasks.** Cluster images into `n_clusters` clusters. Plot centers of clusters as images.
   - **Outputs.** Return the following evaluation metrics: *Adjusted Rand index*, *Mutual Information based scores*, *V-measure*, and *Silhouette Coefficient*.

4. **What is the best number of clusters**? (10%)
   Plot scores of all four evaluation metrics as functions of `n_clusters`. In `description4.txt`, briefly describe the figure you see. Feel free to state your insights about what you see.

[4] Section 2.3.9 in
`http://scikit-learn.org/stable/modules/clustering.html#clustering-performance-evaluation`

Note: You need to submit `ex4.py` and `description4.txt` (`or description4.pdf`) for task 3.2. Your `ex4.py` script should output values of all four evaluation metrics. It should also plot centers of clusters and scores of all four evaluation metrics as functions of `n_clusters`.

4 Submission

Instructions for the submission are as follows. **Please follow them carefully.**

1. Make sure you have answered all questions in your report.
2. Test all your Python scripts before submission. Any script that has syntax error will not be marked.
3. Zip all the required files into a single zipped file named `<student-id>_asgn3.zip`, where `<student-id>` should be replaced with your own student ID. e.g., `1155012345_asgn3.zip`. Please do not change the filenames of the python scripts. The following is the checklist for you:
   - `ex1.py`
   - `ex2.py`
   - `ex3.py`
   - `ex4.py`
   - `eigenimages.jpg`
   - `digit_pov.jpg`
   - `description2.txt` (`or description2.pdf`)
   - `description3.txt` (`or description3.pdf`)
   - `description4.txt` (`or description4.pdf`)
4. Submit the zipped file `<student-id>_asgn3.zip` via CUHK Blackboard System no later than 23:59 on Wed. Apr. 24, 2019.