Assignment 2 – CSC4007 Advanced Machine Learning

Note: This assignment amounts to 30% of the module mark.
Deadline: 23:55 Sunday, 7th April 2019

Summary: In this assignment, you will use k-means clustering to cluster a data set of unlabeled data and also explore some of its applications. Your tasks consist of:

- Using k-means clustering to cluster a dataset of hand-written letters.
- Using cross-validation to select hyper-parameters
- Applying k-means clustering for tuning the RBF features used in Ridge regression
- Applying k-means clustering for image segmentation

All codes are required to be written using Python on Jupyter Notebook. Only basic Python code, e.g. numpy and matplotlib, is allowed. For all tasks, Euclidean distance should be used to compute the distance/difference between data points. **NOTE**: For all tasks (**except task 1.1**), set the random seed to 2019. Task 1.1 requires you to discuss the sensitivity of k-means clustering due to random initialization.

Task 1 (13%): K-means clustering. In this task, you will cluster an image data given in file "*letters.zip*". This is a small subset of the hand-written letters and digits (reference: The EMNIST Dataset). This compressed file consists of 1800 images of 9 different hand-written letters (without labels). Each image file is an RGB image with the size of 128*128*3.

1.1 (2%): Data construction. Unzip the data in file *"letters.zip*" and load all images. For each image loaded, its data structure should look like: image = 128*128*3, let's use only the "R" information for doing k-means clustering, i.e. **extracting image[:,:,0] then flatten it**. Note that the dataset constructed will be a matrix of 1800*16384 (after extracting and flattening).

1.2 (8%): Clustering. Divide the dataset constructed in 1.1 into a training (80%) and a testing set (20%). Use k-means clustering to find 9 clusters for the training set. Report: 1) the SSE error for both the training and testing sets, 2) plotting the SSE error during training vs. iterations of k-means, 3) drawing the 9 image centroids found by your algorithm, and 4) for each centroid, drawing its closest image in the training set. [Hints: Assign each centroid to a random image at the initialization step in k-means clustering].

1.3 (3%): Cross-validation with k-means clustering. Use all data constructed in 1.1 for 3-fold cross-validation to select the number of centroids among the candidates {6, 9, 12, 15}.

Task 2 (15%): An application of k-means clustering. This task is supervised learning. As you have experienced in assignment 1, the performance of ridge regression relies on the quality of features. In this task, you will use k-means clustering to create a set of radial basis function features (RBF) for ridge regression (this is considered as a pre-processing step for ridge regression). You are given a dataset in file '*regression.data*'. The first two columns are input (x), the last column is outputs (y) .

2.1 (7%): Using k-means clustering to find three centroids and assigning them as the set of centers $C=\{c_j\}_{j=1}^{3}$. Construct the set of RBF features for an input x based on $C=\{c_j\}_{j=1}^{3}$ as:

$$\phi_j(x)=\exp\left(\frac{-\|x-c_j\|^2}{2\sigma^2}\right), \forall\, j=[1,2,3]$$

where we denote $\|x-c_j\|$ as the length/magnitude of the vector $(x-c_j)$. Divide the dataset into a training set (80%) and testing set (20%). Use ridge regression (set $\lambda=0.1$) with the above constructed RBF features (with $\sigma^2=2.0$) to fit a non-linear model on the training set, and report the mean square error on both the training and testing sets.

2.2 (4%): Using 5-fold-cross-validation to select a better number of centroids *k* on the range $k=10*i$ where $i=1,2,\ldots,10$ (with other settings similar to sub-task 2.1) (using the training set constructed in 2.1). Report the best *k* and the testing error of its associated optimum model (evaluate this best model using the testing dataset received in sub-task 2.1). Visualize: 1) the curves of training and validation mean square error estimations (over 5 folds) together with their variances and 2) the curves of training and validation SSE error estimations (over 5 folds) together with their variances. Discuss and explain the difference of the results of the cross-validated accuracies on both training and testing data for the models received in this sub-task and the one in 2.1.

2.3 (4%): Using 5-fold-cross-validation to select a better σ^2 for the RBF features on the range $\sigma^2=0.5*i$ where $i=1,2,\ldots,20$ (with other settings similar to sub-task 2.1) (using the training set constructed in 2.1). Report the best σ^2 and the testing error of its associated optimum model (evaluate this best model using the testing dataset received in sub-task 2.1). Visualize the curves of training and validation mean square error estimations (over 5 folds) together with their variances. Discuss and explain the difference of the results of the cross-validated accuracies on both training and testing data for the models received in this sub-task and the ones in 2.1 and 2.2.

Task 3 (2%): An application of k-means clustering. This task shows an application of k-means clustering for image segmentation. The image you will do segmentation is stored in file *"helen.jpg"*. Use k-means clustering to segment this image with 4 centroids, draw 4 image centroids.

Assessment Criteria

The report will be weighted 70%. The evaluation of the report will be based on its quality, clarity, accuracy, and completeness. Use your own words to explain and interpret your methods used for implementation and the results received in each sub-task. You should also demonstrate your understanding and reasoning in every sub-task. For example, what are the results you receive? what is the meaning and interpretation of each result? Why and what are the results different from sub-task to sub-task (as asked explicitly in some sub-task)?

Code (submitted in a jupyter notebook file) will be weighted 30%. Your code should be efficient, concise and has good structure and organization, e.g. try to use functions for repetitive codes, avoid hand-coded settings for variables, etc.. Use comments and markdown cells to explain your code's functionality (can be concise like our code samples for practical labs). All vector and matrix operations must use Numpy. All visualization tasks must use Matplotlib. You are not allowed to use black-box or off-the-self machine learning libraries to solve these tasks.

Guidelines and Submission

Submission including your code (jupyter notebook file) and report must be submitted online, using the QOL webpage of the Advanced Machine Learning module, by **23:55 Sunday, 7th April2019**.

Note that your report should not contain details of code implementation, use code comments and markdown cells in Jupyter Notebook for code explanation. The report should have information (student name, student number, time, data, and assignment number) clearly written in the first page.

All files must be in a ZIP file called “Assignment2_StudentNumber.zip”. It’s noted that your submission is your own work and you are aware of the University policies regarding plagiarism and collusion.

Late submissions will be treated according to standard university penalties.