THE HONG KONG UNIVERSITY OF SCIENCE & TECHNOLOGY
Department of Computer Science and Engineering
COMP4211: Machine Learning
Fall 2020 Assignment 2
Due time and date: 11:59 pm, Nov 22 (Sun), 2020.

IMPORTANT NOTES

- **Your grade will be based on the correctness and clarity.**
- **Late submission: 25 marks will be deducted for every 24 hours after the deadline.**
- **ZERO-Tolerance on Plagiarism: All involved parties will get zero mark.**

1 Tasks

Task1

- You are given the data set data.npz, which contains $N = 1001$ 2048-dimensional points. Load the data set using `numpy`.
- Run PCA (in `sklearn`) to reduce the dimensionality of the data from 2048 to d, such that d is as small as possible while at least 90% of the data variance is preserved in the projected space. Print the value of d you used.
- All subsequent tasks will be based on this data set obtained from PCA.

Task2

- Implement the k-means clustering algorithm using `numpy`. The k-means algorithm should not loop more than 100 times if it does not converge. It should have the following syntax:

  kmeans(dataset, metric, k)

  Here, dataset is a $N \times d$ matrix storing N d-dimensional points, metric is a callable object which inputs two points and outputs the distance between them (see sklearn.metrics.pairwise_distances). kmeans should output the cluster ids of the N points as a vector. Since `sklearn` has already implemented the k-means clustering algorithm, you **SHOULD NOT** use KMeans in `sklearn` for this task, but you can use other utilities (like pairwise_distance).

Task3

- There are a number of *cluster validity indices* that can be used to validate the clustering results. In this task, you have to implement two of these. As these are also implemented in `sklearn`, you have to do your own implementation and **SHOULD NOT** use silhouette_score nor davies_bouldin_score in `sklearn`, but you can use other utilities (like pairwise_distance).

  The first index is the Davies–Bouldin index, which is defined as:

  $$DB(C) = \frac{1}{k} \sum_{C_k \in C} \max_{C_l \in C \setminus C_k} \left\{ \frac{S(C_k) + S(C_l)}{d(\bar{C}_k, \bar{C}_l)} \right\},$$

  where C is the set of clusters obtained, $C_k \in C$ is the kth cluster, $\bar{C}_k$ is the center of cluster C_k, $d(\cdot, \cdot)$ is the Euclidean distance, $S(C_k) = \frac{1}{|C_k|} \sum_{x_i \in C_k} d(x_i, \bar{C}_k)$, and $|C_k|$ is the number of points in cluster C_k. The lower the Davies–Bouldin index, the better is the clustering solution. Implement it in the following syntax:

  db(dataset, labels)

where dataset is a $N \times d$ matrix containing the N d-dimmensional points, and labels is a N dimension vector indicating the cluster id of each data point.

- The second one is the Silhouette index, which is defined as:

$$Sil(C) = \frac{1}{N} \sum_{C_k \in C} \sum_{x_i \in C_k} \frac{b(x_i, C_k) - a(x_i, C_k)}{\max\{a(x_i, C_k), b(x_i, C_k)\}},$$

where

$$\begin{aligned} a(x_i, C_k) &= \frac{1}{|C_k| - 1} \sum_{x_j \in C_k \setminus \{x_i\}} d(x_i, x_j), \\ b(x_i, C_k) &= \min_{C_l \in C \setminus C_k} \left\{ \frac{1}{|C_l|} \sum_{x_j \in C_l} d(x_i, x_j) \right\}. \end{aligned}$$

The higher the Silhouette index, the better is the clustering solution. Implement it in the following syntax:

sil(dataset, labels)

Task4

- Run the k-means clustering algorithm (implemented in Task2) with $k = 3$ on data.
- Output the Davies-Boudldin and Silhouette indices (implemented in Task3) computed on the clustering solution.
- Visualize the data set in 2D by using t-SNE (t-distributed stochastic neighbor embedding) from `sklearn`(function TSNE). Show the obtained clusters with different colors.

Task5

- Recall that k-means clustering requires using a distance between samples. In this task, you are given a blackbox which outputs the similarity $s(x_1, x_2)$ for any pair of samples x_1, x_2. This similarity is always between 0 and 1, and $s(x, x) = 1$ for any x. Show in the report how you can derive a distance measure from this similarity blackbox. Note that if $s(x_1, x_2) = x_1' x_2$ (the inner product), the distance you derived should become the standard Euclidean distance.
- The similarity blackbox can be imported by the statement

from comp4211 import sim

Make sure the provided .so and .pyd files (can be downloaded from the course webpage) are in the same directory as your .ipynb file. The blackbox can be assessed by calling sim(x1, x2, sigma), where x1, x2 are two sample points, and sigma is a parameter of the similarity measure. Based on the distance you derived above, implement a function to pass as the metric in the kmeans function.

Task6

- Perform Task4 again but with the similarity-induced distance measure from Task5. Choose a suitable value for sigma (its default is 0.02). Your grade will be partially based on the values of the Davies-Boudldin and Silhouette indices obtained.

NOTE: You can use *markdown* (covered in tutorial T1) to write a brief description for each task. Please vectorize your code so that it runs faster.

2 Submission Guidelines

- Note: You should use a `python` notebook for this assignment. Notice that you CANNOT import packages other than `numpy`, `sklearn`, and `matplotlib`.
- Please zip your Python notebook (the `ipynb` file) with *markdown* summary, and outputs from all the tasks. For the derivation of the distance (from the similarity blackbox), you can either put it in a `pdf` (and submit also the `pdf`) or include the derivation in the *markdown* (using, e.g., MathJax).
- Please submit the assignment2 by uploading the compressed file to Canvas. Note that the submission should be legible, otherwise you may lose some points if the assignment is difficult to read. Plagiarism will lead to zero points.