CS5284 Graph Machine Learning

Coding sample

Instructions

Name: Please, add your name here : e.g. JOHN SMITH

Answers: Please write your answers directly in this notebook by completing the code sections marked with

`# YOUR CODE STARTS HERE`

`# YOUR CODE` (it can span one or multiple lines)

`# YOUR CODE ENDS HERE` .

Remark: If certain conditions of the questions (for eg. hyperparameter values) are not stated, you are free to choose anything you want.

Exercise : Graph Clustering with World Happiness Dataset

Import libraries and utility functions

In [ ]:
```python
%reset -f
import datetime
print('Timestamp:',datetime.datetime.now().strftime("%y-%m-%d--%H-%M-%S"))
import numpy as np
import scipy.sparse
import pandas as pd
import sklearn.metrics.pairwise
import networkx as nx
import sys; sys.path.insert(0, 'lib/')
from lib.utils import compute_ncut
from lib.utils import interactive_vis_graph
from lib.utils import print_neighboring_countries
from lib.utils import get_same_label_countries
```

Load and visualize the dataset

In [ ]:
```python
wh = pd.read_csv("datasets/2019.csv") # load the dataset
# Features GDP per capita, Social support, Healthy life expectancy, Freedom to make life choices, Generosity, Perceptions of corruption
data = wh.iloc[:,3:].to_numpy()
print('Number of data points is', data.shape[0], 'and the number of features is', data.shape[1])
print('\nFirst three data points:\n', str(data[:3]))
print('\nData statistics:')
wh.describe()
```

Question 1 : Data normalization

Normalize the dataset `data` so that it has a mean of zero and a standard deviation of one along each data dimension.

After normalization, print both the mean and standard deviation of the normalized dataset.

Hint: You can use the functions `numpy.mean()` and `numpy.std()` for this task.

In [ ]:
```python
##############################
# Normalize the matrix data with zero mean and unit standard deviation
# YOUR CODE STARTS HERE

# YOUR CODE ENDS HERE
##############################
```

Question 2: Construct a k-nearest neighbors (kNN) graph

Follow the instructions below to construct a k-nearest neighbors (kNN) graph from the given dataset.

In [ ]:
```python
def construct_knn_graph(M, k):
    """Construct a graph using KNN
    Args:
        M: Data matrix as ndarray of size [n, d], n being the number of data points, d the number of features
        k: the number of nearest neighbors
    Return:
        A: Adjacency graph as scipy.sparse matrix of size [n, n]
    """

    # `dist_mat` is a n x n matrix with the pairwise distance between the data points
    dist_mat = sklearn.metrics.pairwise.pairwise_distances(M, metric='euclidean', n_jobs=1)
    dist_sorted = dist_mat.copy()
    ##############################
    # Instruction: Compute the n x k `dist_sorted` matrix, where each row contains the top k smallest distances for each data point.
    #              You may use function `numpy.sort()`.
    # YOUR CODE STARTS HERE

    # YOUR CODE ENDS HERE
    ##############################

    # Adjacency matrix values `adj_val`
    sigma2 = np.mean(dist_sorted[:,-1])**2 # graph scale
    adj_val = np.exp(- dist_sorted**2 / sigma2)

    # Compute the n x n sparse adjacency matrix `A`
    n = M.shape[0]
    ##############################
    # Instruction: Compute the row and column indices, `row_idx` and `col_idx`, for each edge in the adjacency matrix.
    #              You may use functions `numpy.arange()` and `numpy.argsort()` and `numpy.reshape()`.
    # YOUR CODE STARTS HERE

    # YOUR CODE ENDS HERE
    ##############################
    data = adj_val.reshape(n*k)
    A = scipy.sparse.csr_matrix((data, (row_idx, col_idx)), shape=(n, n))

    # Make A symmetric
    bigger = A.T > A
    A = A - A.multiply(bigger) + A.T.multiply(bigger)

    return A
```

In [ ]:
```python
# Construct the graph
A = construct_knn_graph(data, k=3)

# Visualize the graph of countries
countries = wh.iloc[:,1].values
labels = dict(zip(range(data.shape[0]), countries))
```

```python
G_nx = nx.from_scipy_sparse_array(A)
G_nx.remove_edges_from(nx.selfloop_edges(G_nx)) # Remove self-loops
interactive_vis_graph(G_nx, countries, np.ones(data.shape[0]))
print_neighboring_countries(G_nx, countries, 'Finland')
print_neighboring_countries(G_nx, countries, 'Thailand')
print_neighboring_countries(G_nx, countries, 'Singapore')
```

Question 3: Compute and visualize clusters of similar countries according to the features

Identify at least three countries that belong to the same cluster as Singapore.

Hint: Use the `get_same_label_countries(countries, C, 'Singapore')` function to display the countries in the same cluster as Singapore.

In [ ]:
```python
C, _ = compute_ncut(A, np.ones(data.shape[0]), 20)
interactive_vis_graph(G_nx, countries, C)
##############################
# YOUR CODE STARTS HERE

# YOUR CODE ENDS HERE
##############################
```

End of coding test

In [ ]:
```python

```