Homework 9

Name:

Student ID:

- Reasoning and work must be shown to gain partial/full credit
- Please include the cover-page on your homework PDF with your name and student ID. Failure of doing so is considered bad citizenship.

1. (2 points) **Revise final project**: Start a preliminary document, using the template IEEE Conference on overleaf for your final project. Update the plan, breaking it in tasks and documenting it in the two column paper as follows:

   Abstract Write a short description in the abstract.

   Introduction Expand with a motivation, state what you want to do again as in the abstract and refer to previous work from the literature.

   Problem formulation Write down a mathematical model and/or description of your problem.

   Proposed Approach In here, you will want to propose what you want to do, broken down in tasks

   Analysis and numerical results The fourth section should be having charts, tables etc. that document numerical results. While you will finish at the end of the course, You should visualize one simple numerical results that shows you are set to succeed in your final project, and that should be illustrative of the preliminary tasks of the proposed approach (e.g. you acquired the data and they make sense).

2. (8 points) **Network embeddings**: In this problem, you will use two types of network embeddings to cluster the nodes of a transportation network. In particular, we will use the driving network of Manhattan, provided in the files *nodes.csv* and *streets.csv*, and shown below. (*The python package OSMnx*[1] *lets you download geospatial data from OpenStreetMap and model, project, visualize, and analyze real-world street networks and any other geospatial geometries*)

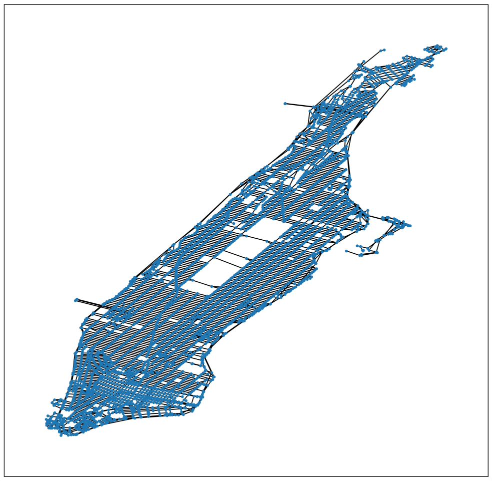

Figure 1: Manhattan driving network

   (a) (1 point) Build the driving network in networkx and use the distance between nodes as the weight of each edge.

   (b) (1 point) Remove the self loops using the function *G.remove_edges_from(nx.selfloop_edges(G))* and plot the network using the coordinates of each node as its position (similar to Fig. 1 but with a different color for the nodes).

   (c) (3 points) **skip2gram approach** Apply the Node2Vec algorithm to the graph (You may rely on the python implementation Node2Vec[2]) (**Hint**: You may want to use the following hyperparameters *dimensions=32*, *walk_length=50*, *num_walks=300*,*window=10*, *min_count=1*, *batch_words=4*. Also, Node2Vec will not return the embeddings in order. Please take this into consideration for the following sections.)

   (d) (1 point) **Matrix factorization** Obtain the eigendecomposition of the weighted Laplacian.

   (e) (1 point) Plot the network embeddings from (c) and (d) in 2D using some dimensionality reduction method (e.g. PCA, TSNE). What do you observe?

   (f) (1 point) Using the results from (c) and (d), cluster the network using using Kmeans. For the number of clusters, use 32 clusters. To validate your results, load the shapefile *manhattan.shp* that contains the neighborhoods of Manhattan, and overlay your results (use geopandas[3] to plot the shapefile). Use a different color for each cluster, and comment your results.

[1] https://osmnx.readthedocs.io/en/stable/

[2] https://github.com/eliorc/node2vec

[3] https://geopandas.org/en/stable/