PPOL 563 - Data Visualization

Vince Egalla

2021/12/11

Final Report

## Executive Summary

Since the beginning of this semester, I’ve been building towards a graph visualization using D3.js. The culmination of this work is a scrolling website, featuring a force-simulated graph, a geographically accurate graph representation of the DC Metro system, and graph machine learning predictions. Prior to this class, my only experience with graphs came from my summer internship; however, developing this web page increased both my understanding of this data type and how to use D3.js for customized visualization. I realized in the end that public transportation systems aren’t an ideal method for graph machine learning, rendering my initial goals unfeasible. I haven’t predicted where a new DC metro station should be located with these methods, but I hope to have built a useful demonstration of this topic.

<https://vegalla.github.io/ppol_563_final/>

## Data

The primary data of this project is a manual creation of graph records from the WMATA metro map.

Example:

|  |  |  |  |
| --- | --- | --- | --- |
| Start | Stop | Line | Color |
| Shady Grove | Rockville | 1 | red |
| Rockville | Twinbrook | 1 | red |
| Twinbrook | White Flint | 1 | red |
| … | … | … | … |

The records reflect connections between two stations, along an arbitrarily assigned line number and the station’s line color. Normally, a graph record explicitly mentions a relationship, but all relationships are consistent for this graph of “X is connected to Y along this line color.”

For the geographic representation of the graph, I use coordinates from the Intermodal Passenger Connectivity Database (IPCD), published by the US Department of Transportation (DOT) Bureau of Transportation Statistics (BTS).

<https://github.com/vegalla/vegalla.github.io/tree/master/ppol_563_final/data/raw>

## Methods

For visualization, I am using D3.js to draw the graphs and animations between text sections of the web page. D3.js handles the initialization of the graphs, movement between circles and lines, as well as color changes. Other functions, such as the scrolling, are implemented through JavaScript, particularly functions written by …

Visualization code: <https://github.com/vegalla/vegalla.github.io/blob/master/ppol_563_final/sections.js>

For data processing and machine learning predictions, I use Python and the packages: numpy, pandas, NetworkX, DGL, and torch. NetworkX converts the raw data into a graph object, consisting of a list of nodes and a list of edges (source/target). DGL handles the machine learning using a graph convolutional neural network. Machine learning is used to predict the nodes’ station color while relying on knowing only a few of the nodes’ true colors. Of all the nodes, only 12 are provided labels. The visualization currently uses a simple neural network with two layers, as more sophisticated examples seem to have depreciated since their publishing.

## Highlights

Key highlights/stories your visualization can be used to highlight (talking points)

* What is a graph?
* What is machine learning?
* Is a metro system effective for graph machine learning?