Improving Public Transit Systems through Clustering and Polynomial Regression

George Trieu
Queen's University
g.trieu@queensu.ca

Pavle Ilic
Queen's University
20pi@queensu.ca

Owen Rocchi
Queen's University
19omr6@queensu.ca

Cahal Ng
Queen's University
20cn16@queensu.ca

Duncan Cheng
Queen's University
17jc66@queensu.ca

***Abstract*—Complete this section last. This should be a very concise summary of the entire paper that describes the motivation of your work, the problem you tackled, the methodology you used, the results you achieved, and the significance of your results. Please limit this section to 100-150 words total.**

I. Introduction

Public transit systems around the world vary significantly, fitting the needs of locals and adapting to the local geography. Urban and city planners spend countless hours finding the optimal routes to operate different modes of transit such as busses, trams, subways, and ferries to fulfil the commuting requirements of the public. Often, this process takes years to plan and decades to execute, all while the urban compositions and needs of the public continue to evolve. This results in building transit based on now outdated information. Building transit for the future is more important than ever as population centres continue to grow, and road infrastructure increasingly becomes more congested for personal vehicles. The biggest reason why people end up buying a car instead of taking public transit is because of the inconvenience associated with using the system – a symptom of poor and outdated public transit planning.

A. Motivation

With the amount of public geographical data that is available, artificial intelligence could be applied to these large datasets to create ideas for urban planners. The main motivation of this project was to create an application that can give realistic solutions to rapid transit lines in any given city.

In this section, using **recent** external references from ***reputable*** sources (typically conference and journal papers), develop an argument as to why you're working on this project/problem. Google Scholar is an excellent search tool. This section should convince the reader that the work you're doing is important, impactful, and relevant. For example, a paper about using machine learning for activity recognition might have the following opening statement:

> With emerging applications in the areas of daily life monitoring and assisted living [1], human activity recognition (HAR) plays an increasingly important role in modern daily life.

Note: When using acronyms, when you first use the term, state the term in proper English, followed by the acronym in brackets (as in above HAR example). From then on, use the acronym. Do not use any acronyms in the title or abstract.

Finish this section by briefly introducing the problem you aim to solve.

B. Related Works

There have been numerous attempts related to our approach to transit route planning, "Improving Transit Accessibility with Machine Learning" by Google AI Blog is one of them, this project uses k-means clustering and regression techniques to predict transit demand and optimize route planning for increased accessibility which gives us inspirations [1]. And there is a guidebook about transit center site selection study to address some good plans in public transit station location settings and amenities important to planning [2].

C. Problem Definition

The objective of the project is to design an optimized subway system for any city in the world using machine learning. Specifically, K-Means Clustering and Regression algorithms will be used to determine the most efficient subway lines. Data sets on points of interests, schools, restaurants and other high traffic locations will be collected to ensure the lines generated are truly the most effective/efficient. The minimum viable product is to generate an optimized transit system for the city of Toronto.

The project will provide a tool for urban and city planners to make informed decisions when designing and implementing a new subway system. It will also give people the opportunity to compare any given subway system with what the most efficient/effective version would be. The final product will be a system that can improve the subway infrastructure of cities, so a greater percentage of the population can be reached.

Overall, the success of this project will be measured by the percentage of people each implemented station serves. This will then be compared to the data provided by the city of Toronto on what percentage of the population the current system services. If the generated system reaches a greater percentage of the population then the project can be deemed a success.

II. Methodology

The datasets obtained were data from OpenStreetMap, via API requests. The desired data was to find all amenities

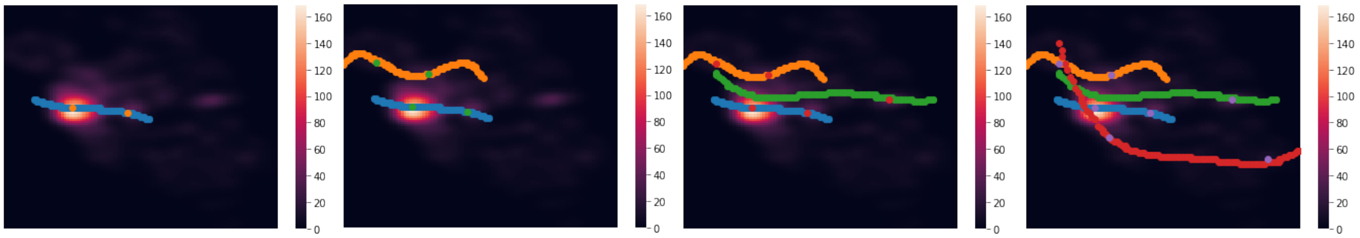

Fig. 1. Line Progression with C = 4 in the City of Toronto

(e.g. community centres, library, restaurants) in a city, and save them into individual CSV files for further analysis. For example, a query would just be a coordinate, which will convert into an Overpass API query, (a read-only API that serves up custom selected parts of the OSM map data) finding the name of the city. Another request is based off the previous city name, then finding all amenities within the city. It then writes the latitude, longitude, and name of each amenity to a CSV file at the given file path and returns a Pandas dataframe containing the latitude and longitude of each amenity.

The result of the query returns a lot of data, with many metadata tags, such as name, date added, other OpenStreetMap data. Most of the extra metadata was filtered out to a table of only the longitude and latitude was returned to a CSV file.

TABLE I
EXAMPLE OF PROCESSED DATA OF RESTAURANTS IN VANCOUVER.

Longitude	Latitude
49.2772564	-123.1187370
49.2630345	-123.1012040
49.2629465	-123.0967703
49.2621476	-123.1007434
49.2567053	-123.1018939

A score function was created to assign every coordinate a measure of usefulness. The score function was based off a Gaussian Distribution as this allowed for locations within a radius of a desired grid point to have a higher score while as the radius between the grid point and location increased the score moved to zero. This scoring function was used alongside the grid search function, which is discussed below, to give scores to each point throughout a city when compared with a data set of the locations of Points of interest, schools, restaurants, etc. Scoring is performed using a Gaussian distribution, defined as

$$f(d) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{d}{\sigma})^2}$$

where d is the 2D euclidean distance between two points, and σ is the standard deviation. This allowed for hyper-parameter tuning as the standard deviation could be altered depending on what radius around a certain grid point should result in the largest scores.

To be able to determine which locations throughout a city are seen as high scoring areas a grid search was performed. This involved breaking the city up into an N x N grid of squares. This grid would then be looped through with the distance of each square and the locations of each point in the data sets being calculated. These distances would be put into the scoring function and the scores for each grid point would be summed up, given a total score. The grid points and associated scores would then be stored in data frame would be stored for later use. A mathematical interpretation of the grid search can be seen below,

$$s_{i,j} = \sum_{k=0}^{N} f(d)$$

where $s_{i,j}$ is score at (i, j), N is the number of points of interest, $f(d)$ is the scoring function. The generated grid points for Toronto is shown below.

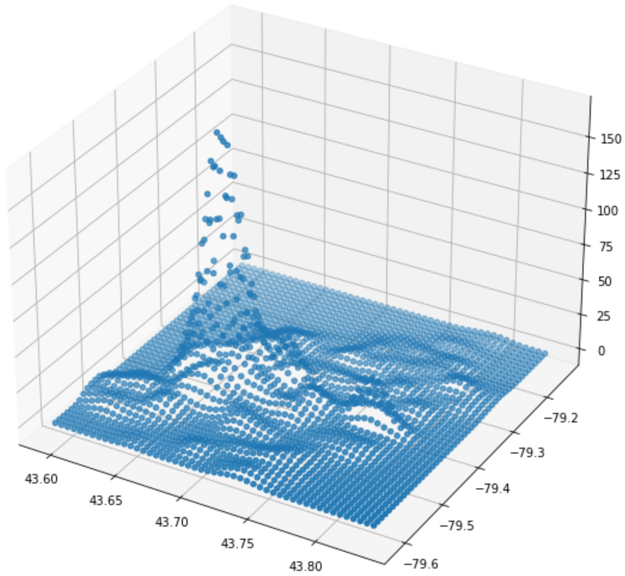

Fig. 2. Scored grid points from the City of Toronto

In any mass urban transportation system, lines in a system should be dispersed throughout the urban area, with additional level of concentration in the densest part (often the downtown area). Dispersing these lines is achieved through clustering. A slightly modified k-means algorithm is used, that considers the scores of each point as a weight w_i.

$$\sum_{i=0}^{n} \min_{\mu_j \in C}(w_i * ||x_i - \mu_j||^2)$$

This weighted k-means algorithm converges on a solution where cluster centers are distributed densely in high score areas, and more sparsely distributed in low score areas. This is synonymous to having more stations and lines in downtown areas, and less stations and lines in suburbs. The clusters represents the neighborhoods each line will go through. This implies the number of lines in the system is equal to the number of clusters. The clusters are numbered at random from 0 to C - 1, where C is the number of clusters. Sorting of these clusters by average score is important for generating interchanges. After sorting, cluster 0 has points that make up the highest average score, cluster 1 has the next highest, etc. The clusters for $C = 6$ in the City of Toronto is shown below.

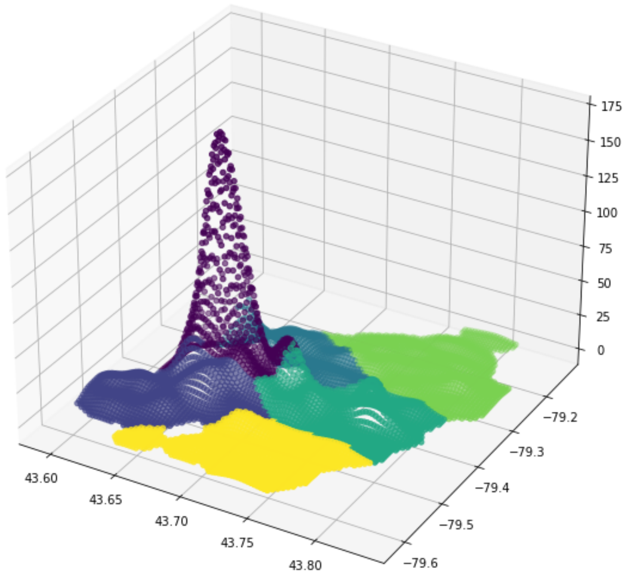

Fig. 3. Scored and clustered grid points from the City of Toronto

After the creations of cluster the problem of fitting subway lines needed to be solved. The solution to this was to isolate each cluster and perform weighted polynomial linear regression on each cluster. Weighted polynomial regression was chosen as it allowed for a prediction of locations based of the scores. Polynomial regression was chosen over linear regression as it allowed for a better coverage of the grid points compared to a straight line. It also allowed for manipulation of the degree of the polynomial to ensure maximum coverage. The points of the regression lines were snapped onto the existing grid to ensure working with the data would be simpler later on.

Interchanges are an important feature in designing mass public transit systems. By running polynomial regression on each cluster separately, the resulting system of lines will not overlap as each line is designed to optimize for its constituent grid points. This implies that there are no interchanges on the system. To address this, an iterative approach to generating lines is considered. Each line is generated successively from previous lines, and utilizes information from previously generated lines. Line generation starts from cluster 0 - the cluster with the highest cumulative score. Potential interchanges, or interchange candidates are identified on each line generation through

$$I_l = \operatorname*{argmax}_{i \in L_l}(s_i)$$

where I_l is the interchange candidate for line l, L_l is the set of points in line l, s_i is the score for the i^{th} point on the line. The value of the grid point associated with I_l is multiplied by a factor A such that the new value V is more favourable to the regression of the next line. This modified grid point is added to the set of points for all future lines, i.e.

$$\forall k \in \{0, ..., C-1\}, L_{l+k} = L_{l+k} \cup \{V\}$$

The interchange candidates also decay after each iteration. This is to encourage future line generation to consider other interchanges and not just the first interchange. After each iteration, each interchange candidate is divided by a decay factor ρ.

To find the locations of each station a function was created that would loop through the points for each one of the subway lines. A station was placed at the end point and from there the distance between that point and the next one was compared to see if they were a certain distance apart. This distance is a parameter passed into the function. If the distance wasn't met the station would be compared with the next point after. If the distance requirement was met then a station would be placed at this location and the comparison process would begin again. This process would continue until all points on the subway line are checked.

The interface the website was built on was React.js. Within the website, the final visualization of the processed data would be displayed in React Leaflet, a JavaScript library that displays interactive maps. A JSON file would be passed into code, iteratively creating lines and stations based off of a general schema.

III. Results

As stated in the problem definition the percentage of the population the final design reaches will be compared to the current transit system for the desired city. By the minimal viable product a comparison will be performed with the city of Toronto.

To determine the percentage of the population the generated subway system reached a function was created that went through each station of each line and calculated the distance between the station and all the grid points from the grid search function. If the distance was less than some max distance then the score would be stored in a separate array for that line. From here the mean score for each line would be calculated.

The mean scores for each line was then compared to the mean score of the each TTC line. The results showed us that insert score maybe add a table

TABLE II
EXAMPLE TABLE SHOWING THE RESULTS OF AN EXPERIMENT.

Model	Accuracy	Precision	Recall	F1
CNN	97.78%	82.32%	88.66%	90.61%
SVM	86.43%	78.41%	67.43%	55.21%
RNN	79.21%	94.13%	80.03%	75.79%

IV. CONCLUSION

- What we've done - Future Improvement

REFERENCES

[1] Wisconsin Department of Transportation. https://wisconsindot.gov/Documents/doing-bus/local-gov/astnce-pgms/transit/ec-site.pdf.

[2] Google AI Blog. "Improving Transit Accessibility with Machine Learning."

[3] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[4] K. Elissa, "Title of paper if known," unpublished.

[5] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.