AI and Itinerary Optimisation

Pierre LAPOLLA
LyRIDS ECE-Paris
Paris, France
pierre.lapolla@edu.ece.Fr

Guilherme MEDEIROS MACHADO
LyRIDS ECE-Paris
Paris, France
gmedeirosmachado@ece.fr

Frédéric RAVAUT
LyRIDS ECE-Paris
Paris, France
frederic.ravaut@ece.fr

***Abstract*—Itinerary recommendation systems are used on a daily basis by a wide range of people, to go from a starting place to their destination. However, most existing systems do not consider accessibility constraints specific to wheelchair users. There is also not much data available on wheelchair users itinerary. In this context, this paper proposes an adaptation and evaluation of an existing itinerary recommendation system: Diffusion Convolutional Recurrent Neural Network (DCRNN). This adaptation is capable to handle multiple locomotion modes, specifically cars and wheelchairs, and computes the time required to travel from point A to point B. Besides the model adaptation, this study proposes a method to derivate wheelchair traffic data from a dataset of car traffic data. Finally, the evaluation of the model is performed through the comparison to a baseline using the Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE) and Mean Squared Error (MSE) error metrics. The source code is available at https://github.com/Deldrel/stage_ing4.**

***Index Terms*—graph neural network, itinerary, wheelchair**

I. Introduction

- Clearly state the problem and its significance
- Introduce the specific challenges faced by wheelchair users regarding itinerary planning.
- Briefly mention the proposed solution, DCRNN adaptation, and its potential impact.

II. Related Work

- Summarize existing research on itinerary recommendation systems.
- Discuss studies focused on accessibility and wheelchair user mobility.
- Highlight the gap in current research that your paper aims to fill.

III. Experiments

A. Model Adaptation

Our approach is heavily based on the DCRNN model [**?**]. This model captures the spatial dependency of traffic data using a custom diffusion convolution operation. The temporal dependency is captured using Gated Recurrent Units (GRU) where matrix multiplications are replaced by the diffusion convolution operation to create a Diffusion Convolution Gated Recurrent Unit (DCGRU). We kept the encoder-decoder architecture of the DCRNN, but we modified some of the layers.

- **DCGRU Cell** : We replace the diffusion convolution operation in the GRU cell by a custom convolution that inherits from the MessagePassing class of the PyTorch Geometric library.
- **Attention Mechanism** : We added an attention mechanism to the model to allow the model to capture the different locomotion modes tendencies.

B. Data Derivation

Due to a massive lack of wheelchair traffic data, we derive it from car traffic data. We use the METR-LA dataset, it contains traffic information collected from loop detectors in the highway of Los Angeles. We have data from 207 sensors, measured between March 1st, 2012 and June 30th, 2012 that is aggregated into 5 minutes windows and normalized using Z-Score normalization. The graph is a weighted adjacency matrix where the weights represent the distance between the sensors, this matrix is build using thresholded Gaussian kernel [**?**].

We use OpenStreetMap [**?**] to get accessibility data for each sensor we have. We look in a 500 meters radius around each sensor every accessibility tag (e.g. *yes, limited, no*) and use (1) to compute an accessibility score for each sensor. Where x is the number of *yes* tags and y the number of *no* tags.

$$f(x,y) = \frac{x}{x+y} \times \log(x+y+1) \tag{1}$$

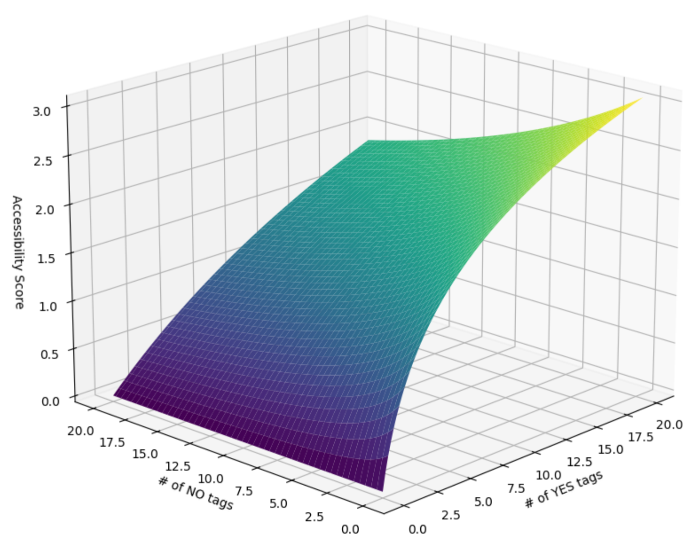

Fig. 1. Accessibility score distribution (1).

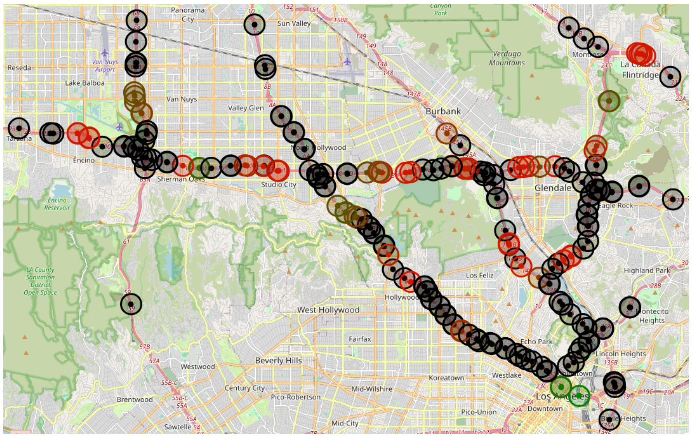

Fig. 2. Map of the sensors with their accessibility score and 500 meters radius around them. The sensors are colored based on their score, red being the lowest and green the highest. Every black sensor indicates that there is no accessibility data available.

We apply a linear transformation to the original traffic data to get values between 0 and 4, where 4 is a value from [**?**]. We add some noise to the data to slightly change the values and make the model more robust. We finally multiply our values by the accessibility score and normalize the data between 0 and 1.

C. Training

The implementation uses PyTorch, PyTorch Geometric, PyTorch Lightning and WandB for monitoring. We split the data into a training(70%), validation(10%) and test(20%) set. The inputs are shaped as follows: (number of samples, number of sensors, sequence length, number of features) e.g. (29977, 207, 12, 4). The sequence length is 12, which corresponds to 1 hour of data and the features are:

- speed (in miles per hour)
- hour of the day
- day of the week
- locomotion mode (hot encoded)

The model is evaluated using the Mean Absolute Error (MAE) metric from PyTorch, defined in (2).

$$f(y_i, \hat{y}_i) = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| \qquad (2)$$

IV. Results

- Present the results of your model's performance compared to the baseline.
- Use tables and figures to illustrate key findings.
- Discuss the significance of your results in the context of the problem.

V. Discussion

- Interpret the results and their implications.
- Discuss any limitations or unexpected findings.
- Suggest areas for future research.

VI. Conclusion

- Summarize the main contributions and findings of your paper.
- Reiterate the significance of addressing accessibility in itinerary planning.
- Highlight the potential impact of your work and suggest future directions.

Acknowledgment

The authors would like to thank Thomas ROUSTAN for their valuable assistance and support during this project.