W19796028 – Stirling Idle

Applied AI Coursework – Part B

AI Techniques in Transportation

Long Travel Times, Traffic Prediction using Dijkstra algorithm, Regression Models & Deep Learning CNNs.

Dijkstra algorithm

Dijkstra is an algorithm which finds the shortest path from a starting node to all other nodes in a weighted graph with non-negative edge weights. The roads are an example of a graph; they all have end points (nodes) and paths which connects roads together. AI can apply the Dijkstra Algorithm to find the shortest path from one destination to another, which can help the driver reduce the amount of time it takes to get from their starting point to the destination. The Dijkstra Algorithm could be used in Path Planning to search for the optimal route for a vehicle and its driver both before and during travelling to their destination, an example of path planning of the shortest path problem would be a vehicle navigation system. Much research has been carried out and have suggested that by combining actual traffic network's distribution characteristic.

The Dijkstra algorithm can be used to for searching the area where the driver is. Because, Dijkstra's algorithm, as well as the spatial distribution feature of the real road network, the algorithm will restrict the searching area to only a sector (it restricts the searching area to a reasonable place). The shortest distance between two points is knows as its displacement, the direction from start point to destination point is generally strike of the shortest path when we plan the route of the real road network. Furthermore, the connection line, which is the line between two points (start and end nodes) and is usually within the vicinity. It says that if there are only one edge between two points, the edge itself is the shortest path. However, in some world, there may exist a path (which is the shortest distance) in the two point's vicinity. In the real world, to enter the right travelled lane, the vehicle will travel the route.

Theortically, the Dijkstra Algorithm is a circle which has a centre S and a radius R, and the theoretical searching area of restricted searching area algorithm is threshold R1. Furthermore, within a circle there is a diagonal line from points S – D, also, if there is a reserve path, each side of the threshold (R1) can be extended outward to a maximum threshold of T2, which as a result forms a larger rectangle which is labelled as R2. The restricted area is calculated by L1 and L2 which cuts the rectangular R2, L1 and L2 which are two lines which are parallel to the line segment Sd and the distance is T. Furthermore, with the increase of road network scale and the distance of neighbor nodes, this means that the difference of time complexity (measures an algorithm's execution time against its input size as it increases). Therefore, the search area of Dijkstra's algorithm reduces the scale of searching, minimizing complexity and will improve the overall efficiency of the system.

Advantages:

- Deterministic and explainable.
- Guaranteed optimal solution – with an admissible heuristic.
- Low computational costs for small to medium networks.

Disadvantages:

- Unable to learn from historical data (regular traffic jam locations).
- There is a performance low when it comes to very large networks.

Input Data:

- Road network which is represented as a graph
- The graph consists of nodes (which are intersections) and edges (which are roads).

Outputs:

- Optimal path from source to destination.
- Estimated travel time or distance.

Deep Learning – Convolutional Neural Networks (CNNs)

Problem: CNNs are usually used for monitoring the volume of traffic and are used in speed cameras to help judge speed of vehicles.

CNN stands for Convolutional Neural Network, and it can be used in image understanding because it has a unique method for extracting critical features from the image which it scans. CNNs have two very important features which make it very useful for image understanding, which are locally connected layers, which is where an output neuron in the layers is connected only to their local nearby input neurons, rather than the entire input neurons in what would be a fully connected layer. This enables efficient image data extraction, because every layer attempts to retrieve a different feature with regards to the prediction problem. The next important component of a CNN for traffic prediction is a pooling mechanism which reduces the number of parameters which are required to train the CNN whilst guaranteeing that the most important features of the images are stored safely and securely. However, a CNN needs to adapt when used in the Transportation sector. It does this in the following ways, firstly, the model inputs are different (the images which are shown to the CNN have only one channel valued by traffic speeds of all roads in a transportation network), the actual pixels themselves have values of 0 to the maximum traffic speed or speed limits (it could use this to judge the speed limit of the vehicle, i.e., speed cameras). The image colour typically has three channels of RGB, because of this the pixels have a range of 0 – 255. Secondly, the model outputs are different, which means in the context of transportation, the model outputs are predicted traffic speeds on all road sections of a transportation network. Then, within transportation, an abstract feature which is extracted by the CNN and pooling layers are relations among road sections with regards to traffic speeds. In an image classification problem, the abstract features could be shallow image edges and deep chapes of some object in terms of its training objective. Then, lastly, the training objectives differ because of distinct model outputs, because of the outputs in which are continuous, the cost functions should be adopted accordingly, as such, in an image processing problem (camera) the cross-entropy cost function is usually used.

Input Data:

- Images/video frames from the sensor of the cameras to the camera feed.
- Classifies data into datasets, including vehicles, lanes and pedestrians.

Output:

- Detected object (Is it a vehicle or person?)
- How congested the road is
- Vehicle speed

Advantages:

- Efficient performance when dealing with image data.
- Automatically extracts features.
- High accuracy for real world traffic and can analyse that traffic data effectively.

Weaknesses:

- Requires large, labelled datasets.
- High computational and training costs.
- Less interpretable than traditional Machine Learning.

Machine Learning – Regression Models

How is regression used in Transportation?

Within the context of transportation, regression is used for traffic flow prediction and travel time estimation.

The article titled "Learning to Estimate the Travel Time" has a section labelled as ABSTRACT which says that vehicle travel time (ETA) is gradually becoming one of the most important "location-based services" and is gradually being used in many new navigation systems and intelligent transportation systems. Firstly, ETA could be thought of as a pure spatial-temporal regression problem which is largely based on a large dataset of effective features. Then, we would use existing machine learning models to solve that regression problem. You could also use a Wide-Deep-Recurrent (WDR) learning model to accurately predict the travel time along a given route at a given departure time. Then, you would jointly train wide linear models, deep neural networks & recurrent neural networks together which will take full advantage of all three models. You could then relate that solution with loads of historical travel data. Furthermore, in recent years, we have seen an increase in how the sharing economy is rapidly changing the life of a large amount of people. A big reason for this is the sharing and online ride-hailing mobile app which redefines the way people move, furthermore, transportation companies such as Uber, Lyft and Didi Chuxing ball of which help people get from one place to another and are seeing a massive benefit on people from day to day. Furthermore, Travel Time estimation is widely used and studied in geographic information systems, there are many solutions, but, the existing solutions can be divided into two categories, the first category is the route - based solution, which represents the travel time using a physical model, where the overall Travel time of a given route is calculated by summing up the travel time through each road segment and the delay time at each intersection, the travel time estimate can be represented as y ˆ = Õ i t ˆ i + Õ j c ˆ j . Furthermore, a road segment or a link between nodes is a route in the road network which has no junctions at the middle but has junctions at both ends, it is the most important part of a road network. Going back to the above formula, the different components are ^ti is the travel time estimated for the i-th road segment, ^C is the delay time estimate at the j-th intersection, this solution has become and is widely used when calculating the estimated travel time.

Input Data:

- Historical traffic data.
- Time of day and the day of the week.
- The overall number of vehicles and their speed as well as, weather conditions.

Output:

- Continuous numerical values – Traffic density (congestion) and travel time.

Advantages:

- Simple to operate.
- Requires relatively small datasets.
- You can train the model fast so that it is able to make predictions fast.

Disadvantages:

- Assumes only linear or simple relationships between data.
- It experiences a very low performance when dealing with complex patterns.
- It has limited output accuracy when compared to deep learning.

What technique am I going to explore in-depth for my prototype implementation in part C?

- I am going to explore the use of the Dijkstra's algorithm in finding the shortest path between two points of a graph and the overall efficiency of the algorithm.