Research Practicum Project Report

Accurate Dublin Bus Travel Time Application

Davy Shaw, Claudia Kinsella, Aongus Bates and Chen Zeng

A thesis submitted in part fulfilment of the degree of

MSc. in Computer Science (Conversion)

Group Number: Group 2

COMP 47360

UCD School of Computer Science

University College Dublin

July 26, 2017

Project Specification

Based on data analysis of historic Dublin bus GPS data, a system which when presented with any bus route, departure time, day of the week, current weather condition, produces an accurate estimate of travel time for the complete route.

Users should be able to interact with the system via a web-based interface which is optimised for mobile devices.

When presented with any bus route, an origin stop and a destination stop, a time, a day of the week, current weather, the system should produce and display via the interface an accurate estimate of travel time for the selected journey.

Abstract

The purpose in creating this application was the design of a accurate travel time application for Dublin Bus. The aim of this was to improve the accuracy of the Dublin Bus service which has occasionally been very unaccurate to the annoyance of its customers. This new application allows the user to plan their journeys ahead of time based upon their inputs into the application.

In our design of this project, we split the project into two sections which we worked on simultaneously: the application itself and the prediction model. The application is a simple web application based upon the framework flask which is used to access the prediction model and the database filled with data relevant to the prediction of travel time. The prediction model is our tool by which we can hope to give an accurate travel time.

In addition to this, we plan to add a number of added features to allows the data to be displayed in a user friendly fashion.

Table of Contents

Chapter 1: **Introduction**

Our main aim when beginning our web application development stage of this project was to ensure that all aspects remained as user-friendly as possible. We have developed our web application using the Flask framework, while using the Jinja template language to ensure the application has a user-friendly design. Python has been our primary language used while developing this application. Static JSON files have been gathered from reliable transport websites and utilised to populate any static data sets used within the application. We populated a SQL database with the cleaned data we had been given, which we used to answer queries that were implemented at a later stage, completed using the Flask framework.

In building the prediction model, we had to look at the vast array of variables that can occur in predicting a journey's tavel time. These included delays due to traffic and adverse weather conditions for example. We began by exploring the different types of models we could use in order to give us an accurate result. We then compared the results of the models and once we were happy with the model we began to scale it up in order to take in more data. In addtion out target feature has changed through the development of this application from a delay to a tiem between stops feature. Once completed, the web application will call the prediction model using variables inputted by the user and return the estimated travel time.

1.1 Description of Problem

At the beginning of our project, as a team, we discussed what we each perceived to be the most efficient and effective way of tackling the proposed problem. Collectively we agreed that our first move should be to understand exactly why a tool like this has not already been developed in a world that relies so heavily on public transport and where saving time and punctuality is of the utmost importance. We came to the conclusion that the majority of reasons behind why accuracy was so hard to grasp in a travel time prediction application was due to the numerous unknown conditions that exist on a day to day basis.

Examples of such conditions consist of the weather, which is a necessity to include in this application, as it changes every day. Public events such as marathons or sporting events occur quite frequently in our society and disturb traffic as roads are closed or blocked off for many hours of the day. One of the most influential unknown conditions is the traffic condition. This has a profound effect on the day to day running of public transport as any minor traffic collision or build up can negatively affect the accuracy of travel time predictions. To overcome this, we have aimed to provide the user of our application with as much information as possible regarding these unknown conditions. Our belief is that if the user is as informed as possible, their decision making process, using our travel time predictions, will allow them to ensure that they reach their final destination at their desired time.

Chapter 2: **Description of Solution/Application**

2.1 Flask Application:

2.1.1 Description of Solution/Application:

We began our approach to building a user-friendly web application by building a skeleton of the application and gradually adhering functional features to it. A brief walk through of how our application works is as follows: a large map of Dublin is displayed with a live weather feed in the top right hand corner and a live traffic condition feed present directly below. Beneath the map is a group of drop-down menus. They consist of the route, origin stop, destination stop, day of the week, time of the day and weather condition. Once the user selects from each menu and clicks the submit button, an area will appear beneath the map which displays the predicted travel time for the selected inputs.

Flask Framework: To initialise Flask, we first established a connection between our SQL database and the Flask framework. Using a config.py file, we provided the details (username, password, port, URI, database name) of our database. From this, we created Flask functions which allowed us to implement specific requests on our data and link these requests to a related URL using the @app.route decorator. Within these functions, connections to the static JSON files were made allowing our static aspects of the application to be populated with information such as the stop numbers and names across Dublin, as well as all the routes and which stops they go to. Using a form within the HTML page, we were able to collect all the data input by the user and store them as variables such as chosenroute. These variables play an integral role as we connect the finalised model with our application using pickle.

Jinja2 Templating: Jinja allowed easy manipulation of our data while trying to populate our application. Using this language, we had the ability to iterate through large sets of data in order to fill the drop down menus present on our application. Jinja has allowed us the flexibility to manipulate the appearance of our application to what we want. For example. Jinja allows us to create a external HTML document which contains the section of code that represents the travel time prediction which is called whenever a user submits a request.

JSON files: We created a number of JSON files from which the user can select their chosen routes, along with the origin and destination of stops they wish to travel from. This was shown to be significantly easier than querying the database directly for this info due to the size of the database and the time it would take for such queries to be made. Once the data has been selected from these files, the relevant variables would be passed to the prediction model in order to return an estimated travel time.

SQLDatabase: The function of this database is to hold the data that the prediction model will use to return an estimated time for the journey. Using the connection we have established between the SQL database and the Flask framework, we have the ability to incorporate SQL queries within all aspects of our application as needed.

APIs: In the development of this application, we implemented the use of a number of APIs in order to enhance the appearance and functionality of our application. The APIs that we used on this project include:

- Open Weather API (To provide real-time weather data to the user)
- Google Maps API (To show routes and stops on a map for the user)

2.2 Prediction Model

2.2.1 Description of Solution/Application:

Historical GPS data allows us to analyse the relationship between travel time and dependent features, such as day of week, hour of day, weather conditions, traffic conditions and so on. However, the raw GPS data was was large and difficult to understand. So in order to clearly analyse the relationship, cleaning data is one of problems we overcame. Different features have different data types, normalization of data was another problem. After data preparation is done, choosing the correct prediction model is the most important issue. Finally, when we built the prediction model and used it in the website, we found that we could not predict the time from place A to B. We could only predict the delay time from A to B, which means that our target feature may not be so accurate. So we changed to a new target feature. For our new target feature, we decided on calculating the time between two stops. However, determining how to calculate it currently remains a problem.

Before doing the prediction models, due to the numeric nature of our target feature, we directly ignored logistic regression. We learnt that regression models from our studies last semester,are much more suited for numeric features. Regression models[1] have been used in many papers about bus travel time prediction by using GPS data and regression models have proven to be outperformed by other models. Because our target feature (delay and time) was of a numeric nature, we still considered other models, such as Artificial Neural Network models(ANN) and Kalman Filter Models(KFM). ANN Models [3] have two approaches to supervised and unsupervised. One of the ANN models is good at classification tasks and the other is excellent at finding the relationships among a complex set of data. In Ranhee Jeong et al [4] journal mentioned that the main reason that ANN Models are considered prediction models for bus travel time is that it can be modified or updated using new data sets. Therefore, in our project, we did not use ANN models to predict bus travel time by using historical data. KFM algorithms work on current data and does not required past information. It applies more to the aerospace field. Certainly, we chose the historical journey model for baseline model to compare with other models, which can help us to see the accuracy.

Cleaning the data: The data provided from Dublin bus was over 3.5GB in size, containing over 40 millions of rows of GPS travel data. The data was assessed thoroughly, which allowed us to understand what data was relevant and what data could be removed. Firstly we assessed the columns. Each row contained 15 columns with the following headings:

- Timestamp, Line ID, Direction, Journey Pattern ID, Time Frame, Vehicle Journey ID, Operator, Congestion, longitude, Latitude, Delay, Block ID, Vehicle ID, Stop ID, At Stop.

The following features were deleted:

- Direction, as this did not relate to the direction the bus was travelling in;
- TimeFrame as this information was already included in Timestamp;

- Operator as the Depot Operating the journeys was not important;
- Longitude and Latitude as our model is based on Bus Stop IDs;
- Block ID as Vehicle Journey ID describes fully the particular bus journey route and scheduled departure times;
- Vehicle ID as the vehicle used for the bus journey was not important.

We then looked at any rows that could be remove. We found that all rows where Journey Pattern ID was equal to null did not correspond to a bus journey, and these rows were removed. We also found that the GPS information provided between bus stops (i.e. en route from one bus stop to another) was irrelevant to our model as no real time data was being received from Dublin Bus. Only the GPS information provided at bus stops was relevant, and so we deleted all the information where At Stop was equal to zero. As a GPS signal was recorded on each bus approximately every 20 seconds, this resulted in us deleting a significant amount of data. The size of the dataframe went from approximately 3.5GB in size, with almost 44,000,000 rows, to approximately 10,000,000 rows of 0.5GB in size.

Solutions:

- Because the raw GPS data was very large and difficult to understand, at the beginning, we just chose one day for one route to analyse what features were very important. So, we deleted features that were less important in the overall raw GPS data.
- After using literature to review different models, we made a decision on using a linear model and a random forest model. Meanwhile, we calculated the accuracy for each model to check which models had the higher accuracy.
- For data normalization, we normalised the day feature in a range from 0 to 6 and a time feature in range from 0 to 23.
- After we decided what features were important and which model we chose, we selected one week of GPS data with weather information to build a prediction model with Delay as our target feature
- For the data preparation section, first approach is that in Chang Wang et al journal[2] had calculated the speed between two stops. Adhering to this approach, we created a new column called time feature which is time between one stop and next stop. So, we changed our target feature from delay to time feature. And we grouped by VehicleJourneyID to calculate the time gap.
- The second method for data preparation was that we can continue to keep delay feature for target feature. But with this method we need to find the real time bus schedule. We can then calculate the travel time by using predicting delay time plus the real time bus schedule.
- In addition to the above, we decided one of our models should be based on historical journey trip times only. By checking how long a particular bus took to get from stop A to stop B on a particular day at a particular time, we can predict the future bus times based almost entirely on this information. For example, if I want to travel from Nassau Street to UCD on the 46A on a Monday at 8:30am, we can see how long this same bus journey took on the recorded 4 Mondays and get an average for our new prediction. This will be used as a baseline model to compare against our main Random Forest model.

Chapter 3: **Progress Report**

The majority of the application has been constructed and allowed the user to input their selections and returns the data to the user. The final step we must complete before completion of our project is to establish a connection between our finalised model and our web application. We are currently implementing this using Pickle.

At the moment, we have built the prediction model by using one week historical gps data. In the further one weeks(24/07/2017 to 31/07/2017), we will use one month GPS data with weather information to build the prediction model. Meanwhile, we will keep our first method choosing delay feature, but we need to find the real time bus schedule. Certainly, we will try change new target feature by calculating the time between two stops, which might be easier to predict travel time. After finishing the two approaches, we will compare the prediction results to see which one is better.

There are a number of added features that we wish to implement before the project comes to a close. These include the addition of some Google Visualization Charts in order to show the user the differences in travel time for their selected route throughout the chosen day and the full week. It also includes the inclusion of social media feeds from Dublin Bus and AA Roadwatch. This can allow the user to monitor traffic conditions that may not have been monitored by the app.

Finally, we are aiming to create what we are calling a proxy cache by which users can quickly access travel times that other users or themselves have frequently accessed. The aim of this would to be significantly cut down the amount of time the application takes to access the prediction model and return a travel time.

Chapter 4: **References**

1 W. Fan, Z, Gurmu. Dynamic Travel Time Prediction Models for Buses Using Only GPS Data. International Journal of Transportation Science and Technology. Vol 4.no. 4. 2015.

2 C, Wang et.al. Real-time bus travel speed estimation model based on bus GPS data.Advances in Mechanical Engineering. Article first published online: November 9, 2016.

3 S, Agatonovic-Kustrin. Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research. Journal of Pharmaceutical and Biomedical Analysis. http://www.sciencedirect.com/science/article/pii/S0731708599002721

4 S.1. J. Chien, Y. Ding, and C. Wei, "Dynamic bus arrival time prediction with artificial neural networks", Journal of Torrsppor.lolion Engineering, ScptemberlOctober2002, p.p.429438.2002.