Traffic Prediction

William Dahl: 001273655
D*ept of Computer Science*
University at Albany, SUNY
Albany, New York, United States
wdahl@albany.edu

***Abstract*—The purpose of this study was to create a system using data mining techniques to predict the congestion of traffic on certain roads in Albany New York at specific times. In this report we will discuss related works, proposed approaches, and our system design and implementation. We found that when working with smaller data sets, it was better to use classification over clustering due to getting meaningless clusters. When concluding the study we saw that having a bigger and more accurate data set was important to having accurate prediction. The used data collection source is twitter and the language used to write the code is Python.**

Keywords—Data Mining, Clustering, Classification, Traffic, Twitter, Prediction, Python

I. Introduction

The motivation of this project is to create an application that could be used by a user to plan there commute around possible daily traffic flow. Traffic is something effects all of us. Whether it is your daily commute to and from work or school or a long road trip, traffic is something that will be there. Hitting traffic can cause you to be late for work or class which can have dire consequences. This application could help the user avoid traffic delays while they plan there commute out ahead of time. The use of this application could also lead to a more even flow of traffic with people leaving at different times to avoid the usual traffic slowdowns. In this project we used twitter as a data source to collect tweets about traffic condition in Albany and used classification to put the tweets into classes based on road and time. Using the size of the classes we then concluded the best time to travel on each major road in Albany.

II. Related Works

A. Google Maps and Waze

As of July 2017, Google implemented a new feature into their app Google Maps. When you search directions on the app a graph will appear with a dashed line that will show the average time your chosen route tends to take within the next couple hours of you traveling on it [1].

Waze, a GPS app known for alerting their users to police or hazards coming up on their current route through other user contribution; and also owned by Google , has added a new feature called “planned drivers” [2]. This feature’s goal is to make it easier to plan your trip based on the traffic you are likely to encounter on the way. In the planned drivers section of the app you can put in where you would like to go and the app will suggest when it is that you should depart are your trip. The app makes these predictions based on “expected traffic conditions based on smart algorithms, aggregated traffic history and predictive analysis." [2].

B. Traffic Prediction by IBM

The Traffic Prediction feature by IBM is apart of there Intelligent Operations for Transportation project [3]. This feature is in most part directed toward authorities; for them to use so they can be there and on the scene where traffic tends to get dense and they can monitor and make sure the rules of the road are up help for the safety of the other drivers. It is also used by Emergency personal so that they can be on standby and ready to react if something were to happen that would require there assistance. IBM uses predictive analytics to do the calculation that the algorithm needs in order to predict the future traffic conditions. Their data set is both past and current traffic conditions collected from the Traffic Awareness system and predicts the future traffic condition in a specific area up to an hour in advanced[3].

III. Outline

The Outline of the rest of paper is as follows:

A. Section 4: Proposed Approaches

Section 4 will discus the many possible approaches that we could have taken for this project. This Section will discuss our options as to where to collect our data from, how to structure that collected data, what data mining technique was appropriate for our task, and how to use our collected data with our chosen technique. Section 4 will also discuss why we chose the data source, data mining technique, and formating that we did.

B. Section 5: System design and implementation

This Section discusses the design for our software System and how we implemented our chosen source for data collection and out chosen data mining technique. Section 5 will outline our mistakes, how we solved them and some of our best solutions. It will discuss the steps taken in our software as well as describe the python files being used and files generated by our code; and the significance and how each generated file is used in the code.

C. Section 6: Results

Section 6 will outline our resulting application that will be seen by the user. This Section will go through how the application is used by the User and how each action carried out by the user is processed by the code. It will display shots of the GUI and the steps the user will take to use the application as well as describe what everything in the GUI means and does.

D. Section 7: Limitations and Challenges

In this section I will talk about the limitations in our application and the challenges that we faced while disining our application.

E. Acknowledgments

F. References

IV. Proposed Approaches

There were many possible approaches that we could have taken for this project. We had to choose where to collect our data from, how to structure that collected data, what data mining technique was appropriate for our task, and how to use our collected data with our chosen technique.

A. Choosing our Data Set

Each Data source that we could have chosen has its pros’s and cons. The data sources that we considered using was Twitter, Cragslist, FourSquare, KDnuggets, and the UCI knowledge Discovery Archive. In this Section we will discuses the different possible sources for data collection and that sources pro’s (a.), con’s (b.) and possible contribution to our project (c.), or what it wont be able to contribute to our project.

1) Twitter

a) Twitter as a data source is great for gathering real-time information.

b) Data gathered could be irrelevant to our project and re-tweets lead to a lot of redundant information. There is also a limit to the amount of data that can be collected through the twitter REST API only being able to get tweets posted in the past 7 days.

c) With Twitter we will be able to use current data collected in real-time that will have a time stamp on it making it useful for predicting times that certain roads will have high traffic levels.

2) Craigslist

a) Craiglist as a data source is great for getting Large amounts of on a large variety of topics.

b) Traffic and news reports are not things that are usually posted to craigslist.

c) For our project the size of postings to craiglist would be great and the ability to get posts that have been up for awhile would be great. However, the lack of traffic reports and news postings remove this from one of the data sources that we could use for our project.

3) FourSquare

a) FourSquare as a data collection source is great for getting a lot of data based on a geographical region.

b) FourSquare is used to find places to eat, drink, shop and visit in a certain city. Traffic and news reports are not something that can be gathered through FourSquare.

c) Although FourSquare supplies a lot of data for a certain geographical region, the data that is supplies does not have to do with traffic and thus is not relevant to our Project, crossing it off of the list for possible data sources to use.

4) KDnuggets

a) KDnuggets is a huge collection of different data sets that can be used for data science and data mining purposes. It also supplies historical data that is extensive and would be useful for predicting traffic trends.

b) The data that can be collected from KDnuggets is not real-time and thus the predicted traffic trends might not be useful or relevant. The data offered by KDnuggets is also not geographically specific so focusing on traffic conditions in Albany would be difficult to do.

c) Looking further into the data sets offered by KDnuggets we couldn't find any data source having to do with traffic. This and the other factors remove it as a possible data source that we could use.

5) UCI Knowledge Discovery

a) Like KDnuggets the UCI Knowledge Discovery Archive is a large collection of Data Sets using historical data which is great for predicting traffic conditions.

b) Again, the data is not geographic centric so focusing on Albany would by difficult. Also the data is not real-time thus, the prediction conditions could be wrong and irrelevant.

c) We also couldn't find any data set on traffic. So UCI Knowledge Discovery was not a source we could collect our data from.

After considering the the different possible data sources, the data source that we chose to use was twitter. Twitter gives us access to real-time information of traffic conditions and allows us to track the time that these reports are posted to that we can predict traffic conditions on certain roads by time. These are things that the other data sources just couldn't provide for us. We specifically implemented a REST API provided by twitter which allowed us to collect data via queries at a range of 7 days.

B. Choosing Our Task

The four main Data Mining Tasks are Clustering, Predictive Modeling, Association Rules, and Anomaly Detections. In this Section I will discuss how each particular Data Mining Task can or can not be used for our purpose with this project.

1) Predictive Modeling

a) Predictive Modeling is creating a model that can predict future events. This would be Great for our purpose because what we are trying to do is predict future traffic levels on certain roads at specific times.

2) Clustering

a) Clustering is used to group together similar data into clusters to help better understand the data and find patterns with n the data. This could work for our Purpose with this project because we could cluster tweets based on road or time, and then cluster within the cluster by time or road respectively. It could however, be difficult to create clusters with in cluster because the amount of data in a certain cluster may not be big enough resulting in clusters that don't really mean anything or display any useful characteristics. Also Clustering on certain characteristics within a tweet could prove difficult because there could be clusters generated based on other characteristics in the tweets that we are not interested in.

3) Association Rule Mining

a) Association Rule mining is when given a set of records, each of which contain some number of items from a given collection; you produce dependency rules which will predict the occurrence of an item based on the occurrences of other items. This could be helpful for our purpose because we could create association rules based around the time and roads with traffic. This however could be difficult because time and roads only go one way. We can only create rules based time pointing to roads or roads pointing to time, not both ways at the same time thus out rules created may not be entirely accurate.

4) Anomaly Detection

a) Anomaly Detection is the detection of significant deviations from normal behavior. This particular Data Mining task is not the most relevant to our purpose in this project. Our collected Data set will contain numerous amounts of data on traffic in Albany and we are just trying to discover the usual flow of traffic on certain road at certain times, not detect anything out of the ordinary.

After considering the four main Data Mining tasks the task that we decided to use was predictive modeling. Considering that out projects purpose is to predict to the user the traffic levels on roads at certain times we felt that

predictive modeling was the correct choice. With predictive modeling we can create a model using the tweets collected to predict the times a road have low or high traffic or which roads will have traffic on them at a certain time.

C. Choosing Our Technique

There are two different approaches for Predictive modeling, Classification and Regression. In this Section I will discuss the differences of each and what each has to offer to out Project.

1) Classification

a) Classification is finding a model for a class attributes as a function of the values of the other attributes. This means that we could classifies tweets based on the time they were posted and the road that is talked about in the tweet. This would create a class within a class and allow us to predict when roads will have high traffic levels at specific times and vice versa.

2) Regression

a) Regression is used to predict a value of a given continuous valued variable based on the values of other variable, assuming a linear or nonlinear model of dependency. This means that you can predict values that are in a continuous field. This isn't really what we are looking for because The values that we are trying to predict are not numbers so they are of a discrete nature as apposed to a continuous nature.

Clearly the choice is classification. With classification we will be able to classify our tweets and then use those classes to predict traffic flow by road and time. Where as with Regression we would have to use number values what just wont fit into our data set and work with the gaol of our project.

In conclusion, we considered many different data sources, data mining tasks, and data mining techniques. After considering the pros and cons of each and weighting what each could provide for us in accomplishing our purpose with this project we decided to use Twitter as a data source because of its ability to get real-time data. We chose Prediction modeling to be our data mining task because it lined up with the over all goal of our project, to predict traffic flow for certain roads in Albany at certain times. And we decide to use classification as our data mining technique because the data that we will be working with is discrete in nature and not continues meaning we can classify our data by road and by time.

V. System Design And Implementation

Our System design included the creation of multiple files, storing our data into different files and formating it differently for reading, and setting up a graphical user interface for the user to use and interact with our application. In this Section I will talk about our System architecture, our datasets, major components and our graphical user interface. Our first step was Data collection, next we classified our data, then we classified the tweets based on road and time.

A. Data Collection

The first step in our system was to collect the data from Twitter. After initializing the Authorization handler and the access token we opened a file called "unlabeled_queries.txt" for writing. This file was to be used for writing all of our retrieved tweets to. We also initialized the geo tag to "42.6529, -73.7562, 50.00mi". This was the coordinators of the city of Albany and with this geo tag all of the tweets that we were mining for were being posted in a 50 mile radius from those coordinates. This made it so that every tweets we were receiving was from Albany.

Next was the development of our queries. Examples of queries that we developed are "Traffic AND Albany", "car accident AND Albany", "traffic AND Washington Ave", and "traffic and Central Ave". We wrote a total of 12 queries which would collect tweets that meet the requirements for each query. The tweets that were collected were loaded in to a list and then we looped through the list of received tweets and wrote each tweets time posted, the user screen name, and the text of the tweet to the file "unlabeled_queries.txt".

B. Data Labeling

After collection out tweets we need to make sure they are actually relevant. At first this will need to be done manually. So, We collected about 1000 tweets and went through each labeled them with either a 1 for positive (relevant and useful) or a 0 for negative. After labeling these tweets we moved them to a text file called "training_set.txt" which is going to be our training set for our SVM, Obviously labeling your entire data set can take forever and just isn't practical. So, we constructed a SVM model to classify all of our collected tweets.

First we wrote a script to collect the top 10 features in all of our collected and manually labeled tweets so far. After we found the 10 most frequently used words in our current data set we began building our training model for the SVM.

Using our current data set as the training set, we created a matrix where each row represented a new tweet and each column represents one of the 10 features. When a feature is found within the tweet the entry in the matrix for the corresponding feature is then set to a 1. We also have a training vector which each entry represents the truth value for each tweet in the file (weather it is a positive or negative tweet – 1 corresponding to positive and 0 corresponding to negative.). After the training matrix and vector is created we create a SVM an fit our training matrix and and training vector to it.

At this point we can re-run our data collection script to collect new tweets that are unlabeled and stored in "unlabeld_queires.txt". We then read through the "unlabeled_queires.txt" file and create a testing matrix and in the same fashion as we did our training matrix. We then use our created SVM to predict the truth values of the tweets and create a predicted truth vector.

Next, for formating purposes, we loop through the predicted truth vector if the predicted value is 1 we then perpend the number 1 to the tweet and write the new line to a file called "predicted_tweets.txt" and 0 if the predicted value was a 0. Then, we loop through the "predicted_tweets.txt" file and where ever the line begins with a 1, it is then written to a file called "quieries.txt" which will contain all of our positive tweets and will be used as out data set for our application.

C. Classifying Tweets

The next step in our system is to classify our tweets by roads and time and make it interactive with the user through a graphical user interface. The first thing we did was set up our abstract data types that will represent the roads and the times. The road data type has the fields count and name.

Count stores the integer count for the number of times the road is found with in the class of tweets. Name holds the string name for the road for when it is displayed in the graphical user interface to the user. The time data time also contains two fields, one called count and another called time. Count, like road holds the count for the amount of times the time appears within the classified tweets. Time holds the string value for the time to be displayed to the user in the graphical user interface.

Next, we create the Graphical user interface (GUI). The module tkinter was used in order to implement the interface. First we created the root window for the GUI which asks the user to chose how they would like to plan their commute. Then, a new window is created which accepts input from the user.

That input is then processed and the corresponding tweets having to do with the entered value are retrieved from the data set. If the user chose to enter a road the tweets that mention that road are retrieved. If they entered a time then the tweets posted at that time are retrieved. The retrieved tweets are then put into classes based on either the time they were posted or the roads that they mention. As the tweets are added to their classes the counter for that class is increased.

Then, Based on the class size, the out put generated and shown to the user is in order form least likely to have traffic to most likely to have traffic.

VI. Result and Analysis

The result after implementing our data source, data mining task, and data mining technique is an application that a user can use to plan their commute. The application allows the user to choose how they would like to pan their commute, by time or by road. When they select one of these options they enter in the time or road they would like to leave at or use and then the option best for them is then generated and displayed to the user. In this section I will discuss what the user will see and the options that user will have to choose from and how the code processes these actions.

When the user first starts up the application they will be presented with a window titled “Plan Your Commute”. In this window they will be Given three options: “By Time”, “By Road”, and “Exit”. Each of these options are buttons that carries out an operation through a function call in the code when the user clicks one of them. When the user clicks “Exit” the program simply terminates.

The roads that we used to classify on are “I-90”, “I-87”, “I-787”, “Washington Ave”, Central Ave, Madison Ave, Western Ave, and the Northway.

The operations carried out by the options “By Time” and “By Road” are as follows:

A. By Time

When the user selects the “By Time” option by clicking on the button, a function call is made to the function called “enter_time()”. This function creates the window for the user to enter the hour they would like to leave. After the user enters the hour they would like to leave they have to press the “Enter” button. Once they press this button a function call is made to the function “get_time()”.

The function “get_time()” just processes the time that the user entered and saves it into a global variable for later use in gathering all of the tweets posted at that hour. After the time is processed and saved the function “time_class()” is called.

The “time_class()” function is where we begin to make our classes based on the roads mentioned in the tweets that are in our data set and were posted in the hour the user specified. The first ting done in the “time_class” is initializing the road data types for what we are going to use. After the road variables are created we then read through our data set a retrieved all of the tweets that were posted at the hour the user provided and the retrieved tweets are stored into a list. We then loop through the list of retrieved tweets and put each tweet into a class based on the road that they mention. As the tweets are added to their classes a counter for each class is incremented. We then put each class into a list and then sort the list by the size of each class. At the end of “time_class()” we make a function call too “print_routes)” which takes in out sorted list of our classified tweets as an argument.

The “print_routes()” function creates a window titled “Best Routes to Take in Order”. It then prints the list of classes and the corresponding name for each road that the tweet was classified into. At the bottom of the window the user is presented three buttons. One is to enter a new time which then calls the function “enter_time()” again and starts the same process over again. Another option is “Exit” which terminates the program. The third option is “By Road”. This option will then initiate the “By road” feature of the program.

B. By Road

The “By Road” feature of the application allows for the user to input a road that they would like to travel on and then get back a list of times where the road is least likely to have higher traffic levels. When the button “By Road” is clicked a function call is made to the function “enter_road()”.

The function creates a window for the user to enter the rod they would like to travel on. Once the user enters the road they would like to travel on they click the “Enter” button a function call is made to the function “get_road()”.

The “get_road()” function simply processes the inputed road by the user and stores it into a global variable to be used later on for finding tweets in our data set that mention the road the user has specified. At the end of the function a call to the “road_class()” function is made.

In the “road_class()” function we classify our collected tweets by the times the were posted. The first step is to create our separate time classes using the time data type we made. After all of the time classes are made we loop through our data set and retrieved all of the tweets having to do with the road that the user specified. We put all of those retrieved tweets into a list and then loop through the list and classify each tweet based on the time that they were posted at. As each tweet is added to its class a counter for the size of the class in incremented. After words, each class is put into a list and that list is sorted by the size of each class. A function call to the function “print_times()” is then made and the list containing the time classes is passed in as an argument.

In the “print_times()” function a window is created with the title “The Best Times to use that road are”. Then the string representation for each class in displayed to the user in the order of lowest traffic level to highest traffic level. The User is given three options, To enter a new road which will make a call to “enter_road()” which will start the process again. To enter a time which will call “enter_time()” and start

the “By Time” process. Or the user can click “Exit” which terminates the program.

In conclusion, the application allows for the user to enter either a road or a time to plan their commute. When the user decided to plan their commute by time the enter a time and then get the roads that are best to take at that time. When the user decides to plan their commute by road, they enter a road they would like to travel on and are given the times where that road will have the lowest levels of traffic.

VII. Limitations and Challenges

One of the limitations on our applications is that in order to use a time that is in the afternoon you will need to enter In the time in a 24 hour formate as apposed to a AM and PM format. For example, if you want to use the time 3 PM you would have to enter the time 15.

Another limitation that is in our application is that we were planning on labeling times and roads as good choices and the others as bad depending on the frequency the road or time appeared in the data set, however meeting the deadline did not permit us to create this feature.

One of the challenges that we faced was in decided what type of data technique and task we were going to do. We originally were going to preform clustering and use the size of the clusters to determine traffic levels. However, after creating a the fist group of clusters on our data set we needed to create another set of clusters in each cluster which means we dint have enough data in our data set to get meaningful clusters. After this failure we decided to go with classification and prediction modeling instead because it works better will smaller data sets.

Acknowledgment

I would like to Acknowledge and thank Rikhil Gandhi for his contribution to this project and his help in labeling data and implementing our data mining technique.

References

[1] Bell, Karissa. “New Google Maps Feature Helps You Avoid Traffic Hell.” Mashable, Mashable, 17 July 2017, mashable.com/2017/07/17/google-maps-app-traffic-predictions/#o62O0wnY_Pqk.

[2] Bell, Karissa. “Waze Update Will Predict Traffic Conditions before You Leave.” Mashable, Mashable, 17 Mar. 2016, mashable.com/2016/03/17/waze-planned-drives/#ytFW1A_u_EqS.

[3] “IBM Knowledge Center.” IBM Cognitive Advantage Reports, IBM Corporation, www.ibm.com/support/knowledgecenter/en/SS9HHZ/transport/ov_tp.html.