Anna Selonick: abselonick@gmail.com
David Ryan: dryan@u.northwestern.edu
EECS 349: Machine Learning
Northwestern University
Professor Downey

Predicting SafeRide Wait Times

The Problem

This project attempted to predict the wait time for a SafeRide on a particular night at a particular time. SafeRide is a free car service on Northwestern's campus that runs from 7:00pm to 3:00am seven days a week. SafeRide protects the health and safety of the Northwestern community by providing a safe and reliable travel option both on campus and throughout the local Evanston area. Unfortunately, students must often wait an hour or more before they can get a ride. Long wait times means students cannot rely on SafeRide to quickly escape uncomfortable or unsafe situations. Many times, this deters them from using the service at all. The long wait times often arise due to many students calling at the same time. If calls were distributed more evenly throughout the night, or if SafeRide had more drivers during their busiest hours, the service would be much more effective. By predicting SafeRide wait times throughout the night and finding out what times are less busy, we hope to provide an accurate way to find out when the quietest times for a SafeRide are. Knowing when SafeRide will be less busy could also help the SafeRide board decide how many drivers to have on call each night.

We also looked at which features tend to affect wait times the most and how. The SafeRide executive board could use this information as a tool to decide how many drivers to have on call each night. For example, it is not immediately obvious how bad weather affects SafeRide wait times. It could keep students from going out at all and thus decrease the wait times throughout the night, or it could increase the wait times if students still go out but are more reliant on SafeRides to avoid walking in bad weather. By analyzing these factors, we can develop a set of preferences for the student body that the SafeRide executive board could use to understand when SafeRide will be busiest.

Data Sets

The wait time data was gathered from SafeRide's twitter, @NUSafeRide. In total there were 10,700 tweets to read from the previous 5 years. Approximately 10,000 tweets contained relevant information that indicated the current wait time. We determined the relevance of these tweets through the use of some simple natural language processing. Tweets from @NUSafeRide that are about wait times follow a very simple format, along the lines of "Wait is 45 minutes" or even just "45 mins". If a tweet followed that or other recognized formats, the time was extracted. If the wait time couldn't be recognized, or if it was deemed to be an irrelevant tweet (such as "Happy Holidays!"), then it was not added to the dataset.

Once a tweet was established as useful, the timestamp associated with that tweet was used to find other attributes. Hour of the night, day of the week, etc could be directly calculated from the timestamp, while such as week of the quarter or the quarter itself was compared against data gathered from the Office of the Registrar. Finally, weather for the specified time period was downloaded from WeatherUnderground.

We used six attributes to try and predict our classifier. Initially, we used hour of the night, day of the week, week of the quarter, and the quarter itself. We later added weather events and the minimum recorded temperature for that night. The classifier was the wait time for a SafeRide at given hour. We used Weka to test our dataset with a variety of learners. We also tried different representations of our dataset, both in discrete and continuous results.

The wait time data was gathered from SafeRide's twitter, @NUSafeRide. In total there were 10,700 tweets to read from the previous 5 years. We determined the relevance of these tweets through the use of some simple natural language processing. Tweets from @NUSafeRide that are about wait times follow a very simple format, along the lines of "Wait is 45 minutes" or even just "45 mins". If a tweet followed that or other recognized formats, the time was extracted.

Methods

We processed the data into both categorical and continuous representations as shown in the chart below.

Attribute	Categorical	Continuous
Hour	0,1,2,...,8,9 (0=7pm)	numeric

Weekday	0,1,2,...,5,6 (0=M,1=T,etc.)	0 if Su-We; 1 if Thu-Sat
Week of Quarter	0,1,2,...,9,10	numeric
Quarter	0,1,2 (F,W,S)	0-2
Weather Events	thunderstorm,snow,rain,none	0 if none;1 otherwise
Min Temp	<0,0-25,25-50,50-75,>75	numeric
Wait Time	0-20,20-40,40-60,60-80,booked	numeric (booked=90)

Initial Results and Analysis

The preliminary testing results for predicting Saferide tweets were not very accurate with, however compared to the baseline cross-validated accuracy from ZeroR of 24.55%, the results were still significant. The most significant attribute was hour of the night. It appeared at the top of the J48 decision tree, which had an accuracy 33.69%. This was expected because the initial graphs of the data showed the most drastic variations in wait times across hours of the night. Based on these initial results, it was predicted that adding weather attributes would improve overall accuracy in the predictions. It was also decided that the choice to make all of the features nominal may not have allowed the weighting algorithms to understand the relative difference between each hour of the night or between nights of the week, and this may have negatively affected the results. Thus, in the next tests, two more data sets were created. One data set used discrete nominal attributes and the other included continuous numerical attributes. The continuous numerical data was used to create regressions and compare these results to those of the categorical algorithms.

Updated Results and Analysis

After adding the two attributes for weather to our data set, we also discovered a few different ways to improve the accuracy of our tweet NLP, getting rid of around 300 tweets that were ‘bad’ data. In doing so we were able to improve the accuracy of our results, and moved forward with dividing them up into discrete and continuous representation for further analysis.

Discrete

Overall, the Bayesian algorithms produced the most accurate results. These results were shown to be significant in comparison with the baseline value of the ZeroR Tree that returned an accuracy of 28.3986 %. The best test result from the discrete data was from the HNB algorithm with an accuracy of 39.27% with 359 correctly classified instances. This algorithm weights to the attributes, and this may have contributed to its accuracy. From looking at the data during preprocessing, it appeared some of the attributes had more influence than other on the wait time than others. The wait times did not vary greatly across each day of the week, however it fluctuated greatly depending on the hour of the night. These were merely correlations, but if similar patterns held true for the test data, weighting certain attributes could create a more successful model. The Lazy Bayesian Rule algorithm also worked well. This had an accuracy of 38.29% with 250 correctly classified instances. Naive Bayes classifiers assumes attribute independence, however the LBR algorithm selectively relaxes this assumption. Some of the attributes used were strongly correlated. The two weather attributes: minimum temperature and precipitation were correlated because the low minimum temperatures have a strong correlation with thunderstorms and rain. Additionally, there is a strong correlation between winter quarter and low minimum temperatures.

Continuous

We were able to produce significant results with our continuous data as well. After converting the data set to reflect numerical values, the ZeroR classifier predicted a value of 42.266 and produced a root mean squared error of 26.2974. When evaluated on our test dataset, it was able to place the data into the correct range 24.316% of the time.

Other continuous classifiers were able to perform significantly better on our dataset. The M5P classifier produced the smallest root mean squared error at 24.9658%. Its success may have come from the fact that it uses both decision trees and a series of linear regression equations in order to classify the data. Since some attributes in the data set are better represented as splits rather than a continuous range, creating a preliminary decision tree to separate different examples based on those attributes could have further specified and improved the accuracy of each of the linear regression models.

The KStar attribute was the most accurate when comparing its prediction to the actual value in our discrete ranges (0-20,20-40, etc). It was able to place the prediction in the same range as the actual value 31.05% of the time, and had a root mean squared error of 25.0113. The KStar algorithm uses entropic distance measures in order to classify similar instances, so its success may be attributed to the fact that many nights on SafeRide

may follow a similar pattern. Some of our attributes, such as minimum temperature, likely do not change too much on a night-to-night basis, so it is possible that many nights had very similar values for their attributes. Additionally, the use of an entropy distance measure means it it better suited to handle the mix of real-valued and symbolic attributes that made M5P successful.

Conclusion

Comparing Results of Discrete and Continuous Data

It is clear upon examining the results from the discrete and continuous data sets that the classifiers in Weka were much more suited to handle the dataset when represented symbolically. The continuous classifiers were consistently less accurate, and classifiers used on both data sets such as KStar were outperformed 35.77% to 31.05%. It is likely that this is because the attributes in the data set translate poorly to continuous values. Those that are more naturally continuous (such as hour of the night) were still valuable in dividing up the data when converted to symbolic attributes, as seen in the decision tree for the J48 classifier. It is also significant that the two most successful classifiers for the continuous data set were the KStar and M5P classifiers, as both employ techniques which can handle symbolic attributes well. It is clear that there is room to further explore ways to optimize each of these types of data representation.

Future Research

Weka was useful in the great range of algorithms it allowed us to explore, however it was limiting in the ways that were were able to represent the data. In Weka, in order to use the bayesian and tree algorithms all attributes had to be represented as categorically and nominal, and in order use regression models, all data had to be represented as continuous numerical data. Future research could explore new hybrid representations of data that use both discrete and nominal attributes representation. Optimally, hour of the night should be represented numerically, so that the algorithm can understand the close distance between 11pm and 12am and larger distance between 7pm and 3am. The quarters are less related, and it would be interesting to represent this attribute nominally.

Furthermore, future research might explore representing the weather data differently. In this model, the weather attribute was binary. It was true if there was a thunderstorm or rain and false otherwise. This might be better represented by making an attributes for each type of weather: thunderstorm, rain, snow, and hail. and making each one binary. Finally, the weather data we used could have been more detailed. It would be

good to incorporate an attribute that denotes the weather at each hour of the night. Other weather attributes that could be added include wind, “feels like” temperature, and the temperature relative to the temperatures from the month or the average from previous years. Given the 6% gain in accuracy obtained from adding two weather attributes, adding more detailed weather attributes could increase accuracy even further.