NextBuzz

Nicholas Petosa

Motivation

It is a widely held belief in the Georgia Tech community that the bus system is unreliable. Georgia Tech students over reddit have asserted time and time again that time-to-arrival predictions made by NextBus are inconsistent with actual arrival time [1]. NextBus' unreliability is not unique to Georgia Tech – the SF examiner has published an article highlighting how unreliable NextBus predictions are for San Francisco streetcars, with predictions being inaccurate 40% of the time if a vehicle is 20 minutes away [2]. There are factors that NextBus does not seem to be accounting for in its predictions, like rush hour, weather, and other community-specific variables, like when class lets out. Many students have given up on buses altogether and walk from class to class [1]. They are paying $85 per semester for a service they are not even using because it is so unreliable [3]. Observant students that still use the buses have developed soft rules like "a blue bus at 9PM can do full loops in 10 minutes" in order to accurately forecast bus arrival [1]. The goal of this project is to perform data analysis on historical NextBus data to quantitatively measure its accuracy, and then to build a machine learner which can recognize patterns imperceptible to the casual observer and produce better time-to-arrival predictions. Once the model is built, I will create a simple API and web app which displays improved arrival time predictions relative to the closest stop.

Related Work

NextBus is already in the business of predicting bus arrival times. NextBus makes predictions for buses across the country, not just for the buses at Georgia Tech. As a result, their predictions are lacking domain knowledge specific to local communities, like when rush hour occurs or when classes let out. NextBuzz will build on NextBus' time-to-arrival estimate to produce a more accurate prediction of bus arrival time.

There have been other attempts to deep-dive into the accuracy of NextBus predictions. Tommy Leung developed the NextBus Delay Tracker, a project which uses linear regression on two weeks of Cambridge Massachusetts NextBus data to generate better estimates [4]. The most obvious difference between our projects is that they concern different regions, but the most significant difference is that Tommy did not integrate external factors like weather and class schedule into his predictor. Further, he uses linear regression, a parametric model which will perform sub-optimally on this non-parametric multivariate problem. Another deep-dive into NextBus accuracy was performed in 2015 by the transportation tech company Swiftly. They published a post on medium breaking down how NextBus accuracy plummets as it tries to predict arrivals further out in time [5]. Swiftly does not attempt to use this data to solve prediction accuracy.

There are a number of major players which are currently providing Georgia Tech students and faculty with arrival predictions, and all of them are based on the NextBus API. The most rudimentary of these are the digital signs present above most bust stops around Tech, which display NextBus' predicted time-to-arrival. The problem with this medium is that students who are still in their dorm are unable to access arrival information. To solve this, NextBus provides an website which displays its predictions [6]. The UI for these predictions is clunky, particularly its map for tracking bus location. The best option students have for accessing bus arrival time predictions are the GT Buses apps, available on both the Google Play [7] and iPhone App Store [8]. These mobile apps provide a clean and convenient interface for tracking buses and accessing arrival data. The only problem that remains is that their

predictions are inaccurate, since they are sourced from NextBus. The goal of my web app is not to replace existing solutions, but rather to draw the attention of the GT Buses authors to my API so that they might integrate my accurate predictions into their apps.

Features of Proposed Work

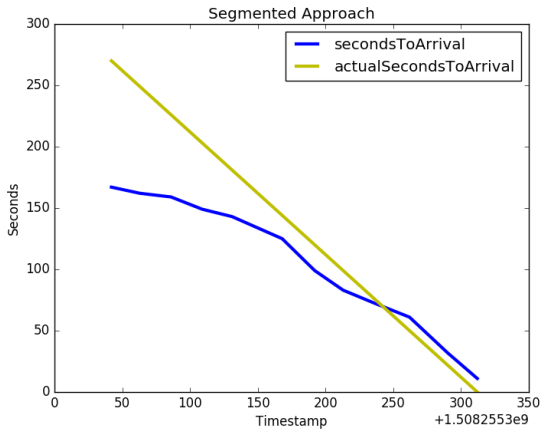
- **Scraper for aggregating NextBus bus data real-time**, including time-to-arrival prediction, bus locations, bus speed, number of buses running, and timestamp. The scraper must also integrate external factors into each row of data such as weather, class schedule, and bus shifts.
- **Parser which can translate raw NextBus data into meaningful structured components.** For example, the NextBus API does not provide any notion of an actual arrival event occurring, or a bus going out of service. These events need to be segmented heuristically, which is a non-trivial problem. This parser will also produce dynamic analysis in the form of graphs and metrics on the provided historical data, such as:
  - Number of days of bus data detected.
  - Number of arrival events detected, broken down by route.
  - Mean absolute error and mean squared error of predictions per route over time. This is the most powerful metric that can be extracted, as it tells us how distance between time-to-arrival and actual arrival impacts prediction error. It will justify the creation of a model that can generate better estimates.
  - Correlations between variable and arrival error, essentially measuring the degree to which each factor predicts error.
  - Performing unsupervised learning to analyze how instances are clustering around each other. Use these clusters to engineer new, more powerful features.
- Use these factors to create **an online machine learning regression model** capable of producing a more accurate time-to-arrival prediction. Retains the most recent month of historical data for training. Graphs how the error of these new predictions compare to NextBus' predictions.
- Host this machine learning model behind **a REST API**, and use that API to build and host a **web app** which displays real-time NextBuzz predictions for the nearest stop.
- Produce a paper documenting the findings of historical data analysis and the effectiveness of my machine learning model in generating smarter predictions.

Evaluation Plan

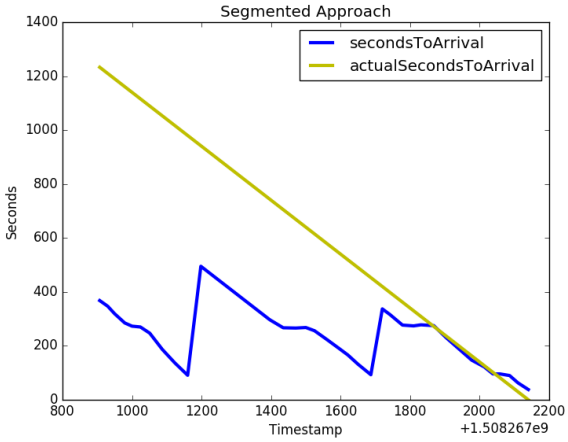
The NextBuzz system relies heavily on its parser to interpret real-world data and its model to predict real-world results. It is crucial that these components be as accurate as possible to maximize the effectiveness of our system, and so we need to have a plan in place for quantifying the accuracy of our parser and model.

Segmentation Heuristic Evaluation – Measuring Fidelity of Segmented Components

The non-deterministic part of our parser is detecting arrival events in a continuous series of bus time-to-arrival data. Arrival events are detected by looking for large and sudden spikes in arrival time, indicating that a bus has just passed that stop. But this is real-world data, and it is difficult to find a "one size fits all" rule for arrival events. For example, red buses arrive more frequently than green buses, so the spike in green bus arrival time is much greater than that of red buses. In order to measure the true positive rate of our heuristic, I will need to draw a large random sample of detected arrivals, and as a human determine what proportion of those segments are correct. To understand what a good case looks like versus what a degenerate case looks like, consider the two segmented components below, which were detected using my current parser.



Good case



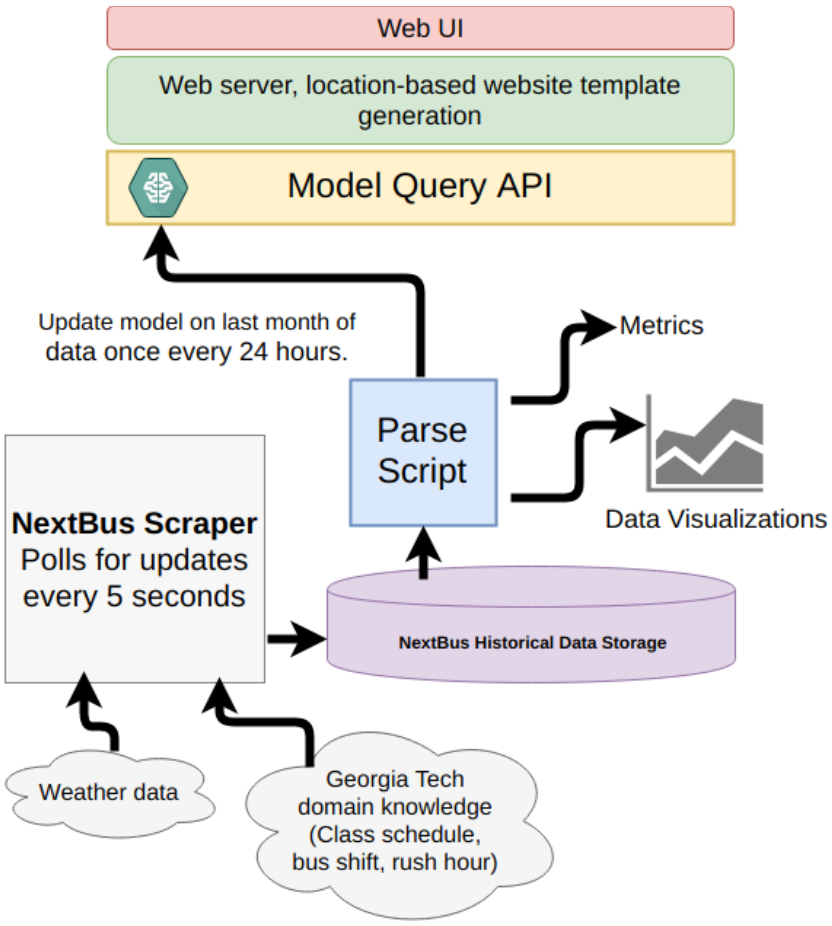
Degenerate case

Notice how in the good case, a bus approach event is completely captured, and actual seconds to arrival is correct. Contrast this with the degenerate case, in which we seem to be missing some arrival events and are miscalculating actual seconds to arrival. By analyzing degenerate cases, we can iterate on our heuristic to more effectively detect arrival events.

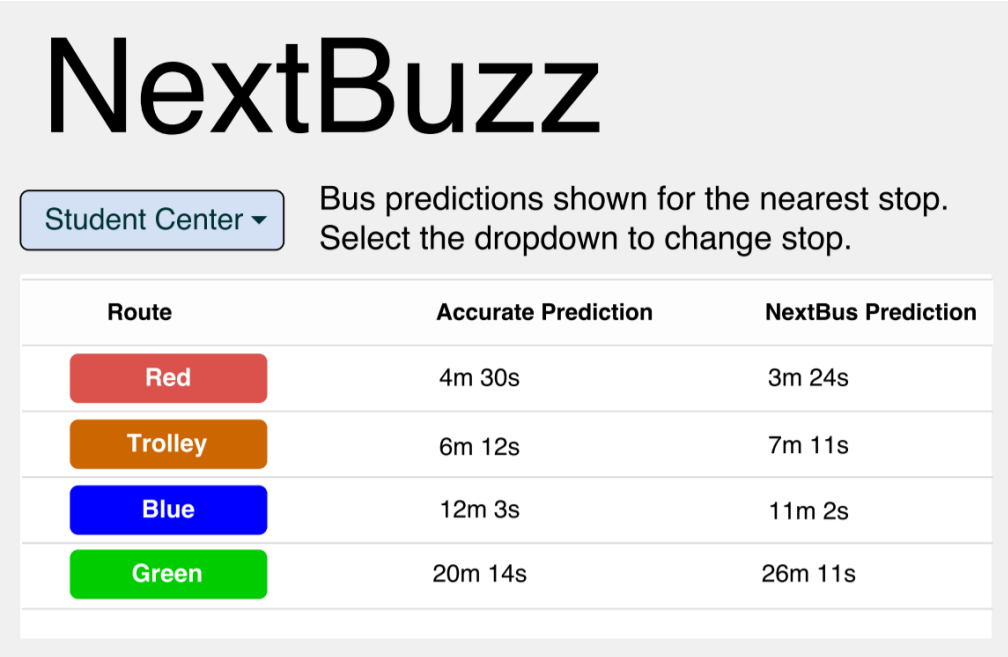
Regression Model Evaluation – Quantifying Prediction Accuracy and Effectiveness

There are two things we need to measure about our model. The first is its accuracy in predicting unseen data. Ideally, error between prediction and actual value is as small as possible. We will use k-fold validation to measure how well a model built on this data can generalize. The other aspect that we need to measure is how much better our model's predictions are compared to NextBus' predictions. This can be shown visually by plotting error versus time for both predictions. Visually, if NextBuzz error is always lower than NextBus error, we know that it outperforms NextBus predictions. We can calculate the percent difference improvement in error by averaging percent change in error between all NextBuzz and NextBus predictions for all data points.

Application Architecture



Application Layout



Project Deliverables

- Working demo of NextBuzz web UI
- Source code for NextBuzz data pipeline and web UI
- Report including NextBus data deep-dive and model performance analysis
- Documentation

Foreseen Risks

- NextBus predictions may be very accurate, obviating the need for this project. This is very low probability, based on community feedback and preliminary results.
- Another risk is that NextBus' error will be completely random, or I will not have enough variables to improve predictions.
- A final risk is that my improved prediction API does not attract the attention of the GT Buses developers, meaning the discoveries made in this project will be low impact.

Planned Schedule

	16-Oct	23-Oct	30-Oct	6-Nov	13-Nov	20-Nov	27-Nov
Build Scraper							
Build Parser/Analysis Generator							
Data engineering to create features							
Tuning model and measuring accuracy							
API/Web UI							
Write report							
Scrape data							
Documentation							

References

[1] https://redd.it/6veyy8
[2] http://www.sfexaminer.com/nextbus-muni-predictions-inaccurate-during-commute-hours-almost-half-the-time-study-says/
[3] http://www.bursar.gatech.edu/student/tuition/Fall_2017/Fall17-all_fees.pdf
[4] https://www.tommyleung.com/nextBus/nextBusBehindTheScenes.htm
[5] https://blog.goswift.ly/san-francisco-transit-prediction-accuracy-how-swyft-helps-you-commute-smarter-ab189cfedd71#.ldum05nhh
[6] https://www.nextbus.com/?a=georgia-tech#!/georgia-tech/
[7] https://play.google.com/store/apps/details?id=me.siddu.betternextbus&hl=en
[8] https://itunes.apple.com/us/app/gt-buses/id815448630?mt=8