Predicting Crime Exposure for Public Transit Takers

Will McFadden

June 11, 2015

Abstract

I have implemented a web interface to display crime predictions for public transit routes in the city of Chicago. I evaluated several models for prediction and chose the one with the lowest false positive rate. The tool provides interesting perspective on the sensitivity of different routes and how time of day matters.

1 Introduction

Both native Chicagoans and visitors to the city will often find themselves traveling on public transit through neighborhoods that can make them feel unsafe. To help prevent crime and to make people feel more safe, I've build a tool that evaluates the likelihood of criminal activity based on transit route and time of day and year.

The problem of predicting crime has been tackled before at the city, state, and federal level. However, any of the academic research I saw on this matter focused on its relation to policing strategies rather than public awareness. Citizens are certainly already using anecdotal accounts of high crime regions to avoid injury and theft. With my application, I'd like to build a consumer tool that allows individuals to use actual data to take a part in protecting themselves against crime day to day.

Since 2000, the city of Chicago has been tracking all of their criminal activity using their CLEAR database. The majority of the data on crime type and location has been put online to allow the public to have open access to it. This has been used in other projects to display local crime statistics.

A specific challenge with my project is that I wish to make some claims about the possibility of predicting future crimes, down to the hour and block. To get help on this problem, I spoke with Maggie King from Brett Goldstein's group at the Harris School of Public Policy. Their team has been working on methods to better predict crime using weather and demography information. I learned a lot about their current methods, and used them as a good baseline for some decisions about my own method.

2 Methods

2.1 Connecting crimes and routes

To begin, I wanted a sample collection of routes to analyze. To do this, I built and deployed a MapReduce algorithm to associate crimes with their corresponding routes on AWS. It works by sending a route to a bin corresponding to a gridded region of Chicago. It then does this with the crimes as well based on their latitude and longitude. The two are combined in a reduce step and then sent to another reduce step whose keys are the routes. This results in a route having all of its crimes associated with it. An earlier version of the app included the capacity to investigate these routes to view the crimes that occurred along them.

2.2 Prediction models

I experimented with 4 classification and 1 regression model to try to predict if a crime would occur in a given spatial block at a specific block of hours in a specific block of months. I experimented with my data binning and determined that the most fine-grained prediction with reasonable accuracy and speed would use half square mile blocks, 3 month windows and 4 hour blocks to aggregate crimes.

Using this method I trained models to predict probability of a crime occurring given a tuple [x,y,m,h], where x and y were the grid longitude and latitude and m and h were the month bin and hour bin. To train I also generated random coordinates and times and assigned them a class of no-crime, varying the number from 3 to 6 times the number of crimes. I used logistic regression, Naive Bayes, SVM, random forest classifier and random forest regressor. Each achieved slightly different results as evaluated with a

ROC curve.

2.3 Normalizing to ridership

Because highly trafficked regions will on average have more crime without actually increasing the risk to an individual, I decided I needed to try to account for local differences in the number of pedestrians for normalization. Because this type of data is not readily available, I instead turned to local bus stop ridership data from the CTA as a proxy for the number of people on the street. The intuition goes that more people getting on and off of buses would correlate strongly with the number of pedestrians passing through an area.

This metric is even more beneficial since the target audience of the application is those people using CTA. Needless to say, the number of people riding the CTA is not strictly proportional to the number of pedestrians on the street at every location in the city so we'll introduce some errors in our estimates. In a perfect world we would have more data to make this assessment better, but as it stands, this is the best data I could find to solve this problem.

2.4 Scoring public transit safety

We can integrate the block-by-block spatial crime prediction with CTA route stop locations to determine a route safety prediction. To do this, I queried the google maps API with a start and end location. The returned route includes stops and walking directions between stops. I chose to focus on just legs of the journey that include significant walking as those would be the places where the crime was most likely.

To develop the scoring parameter, I contemplated two different schemes. The first produced a **integrated** crime likelihood based on the per block crime prediction multiplied by the length of time in that region. This proved to be too difficult to normalize effectively without having a precomputed set of routes so it was discarded. Instead I chose simply to take the **maximum** crime prediction score along the route. The downside is that this gives undue importance to a single stop, but it also allows us to generate our most conservative estimate possible.

2.5 Web interface

Prediction models on spatial data really are generally more useful when the spatial information is put into the context of a map. To do this I built a web interface that integrates the spatial data and predictions with our route scores. The general interface looks like Figure 1.

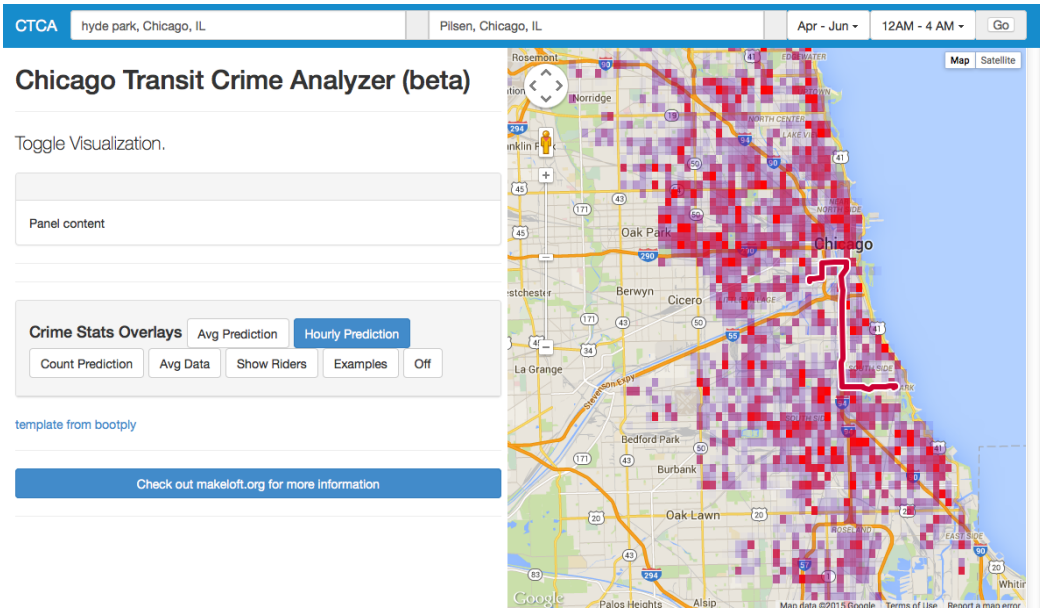


Figure 1: Display interface layout

The beta version of the web interface can be found at
`http://home.uchicago.edu/~wmcfadden/ctac/`

3 Results

3.1 Comparison of models

One important question I had to determine was which model would best be able to make predictions. With very few crimes and very many riders, I recognized that I was bound to run into many false positive in my prediction. Nevertheless I wished to minimize these so I compared different models to see which had the greatest distinguishing power. The ROC curves presented in Figure 2, shows the relationship between the number of false positives and

true positives. The curve for the random forest very clearly wins out by al-

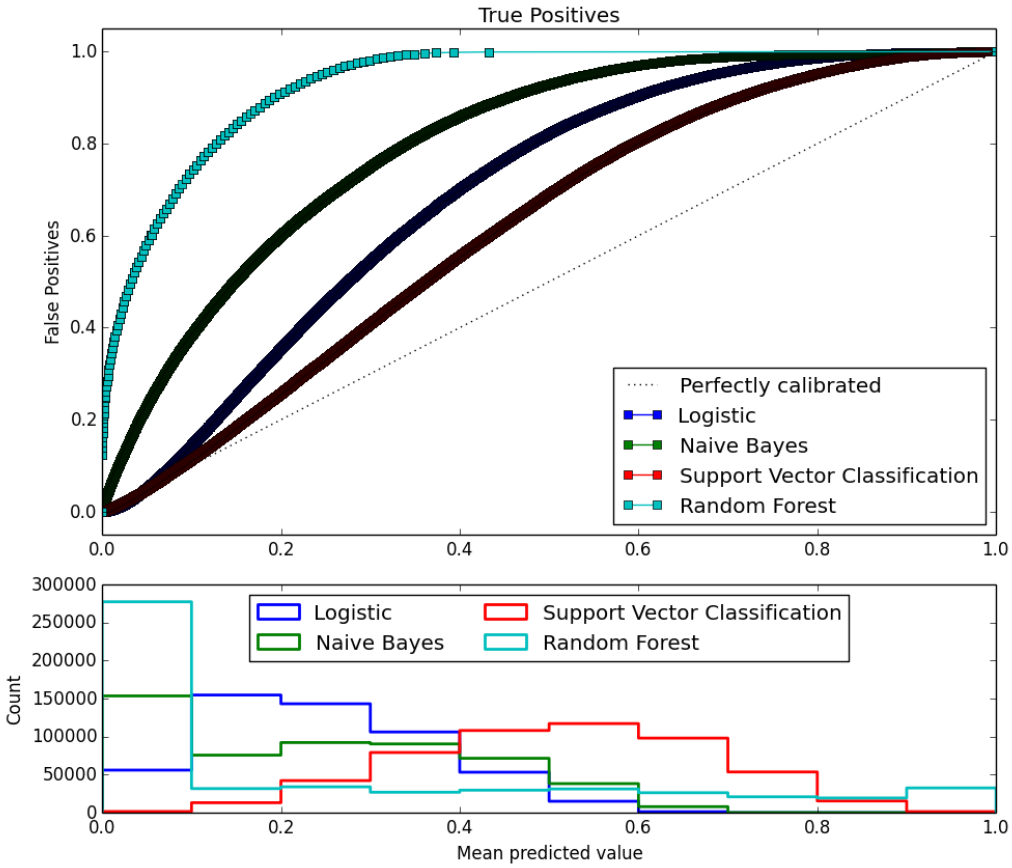
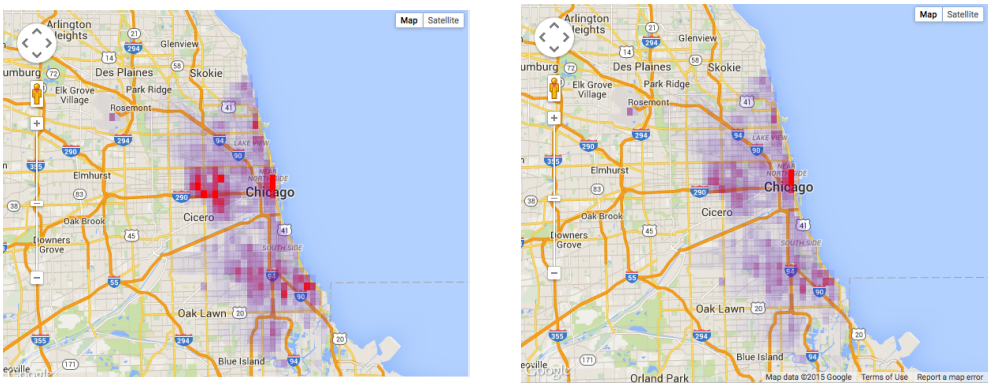


Figure 2: ROC curves, axes labels are reversed

lowing a roughly 90% detection rate with only 20% false positives. This level was comparable with that attained by the Goldstein group. The problem of overestimating is in my opinion less problematic than underestimating.

3.2 Filtering out data

I found that it was very important to filter out which crimes you think are unimportant. In Figure 3 we see the difference in the crime distribution with and without including narcotics violations. There is quite a bit of narcotics only crime in the west side of Chicago. It is important to consider whether



(a) With Narcotics

(b) Without Narcotics

Figure 3: Comparison of the effect of including narcotics offenses in the dataset.

these crimes should worry pedestrians.

Additionally, there was also a slightly humorous bias for one block in the loop. Figure 4 shows the normalized crime distribution depending with or without crimes involving a store.

Interestingly there was a single highly concentrated block of crime directly downtown. It turned out that one block had many hundreds of reported department store thefts. It became clear that crimes at store locations were probably unimportant to crimes against pedestrians so they were removed from the analysis.

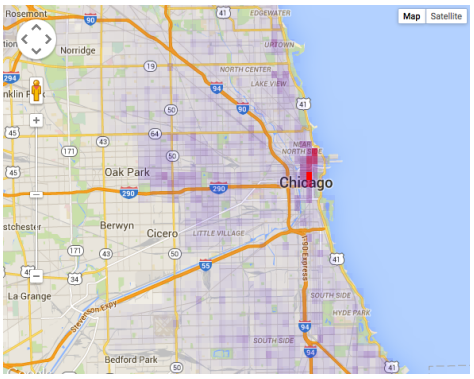
3.3 Unforeseen sensitivity

In the Figure 5, I found that the predicted crime probability could change dramatically with only slight variation in destination.

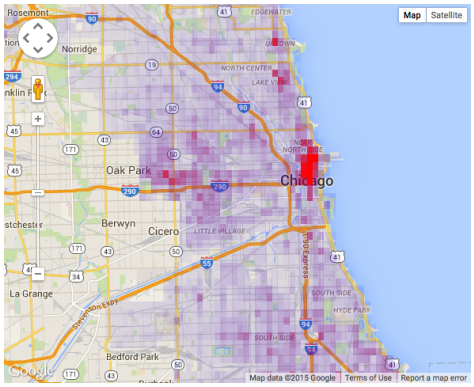
It was interesting that although the end destination only changed slightly in the amount of predicted crime, the route itself. This was, of course, due to the transit route going through different neighborhoods, but it was striking nonetheless.

3.4 Comparing classification and regression

After some thought, I realized that I could do better than just predicting **whether** a crime will occur on a block. It would be better to have a pre-

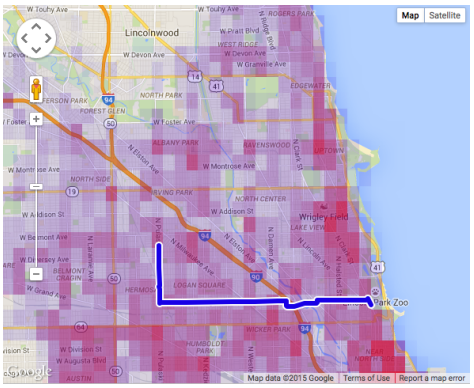


(a) With Crimes in Stores

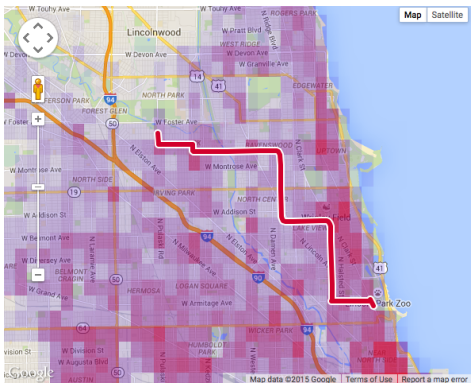


(b) Without Crimes in Stores

Figure 4: Comparison of including crimes in stores in the dataset.

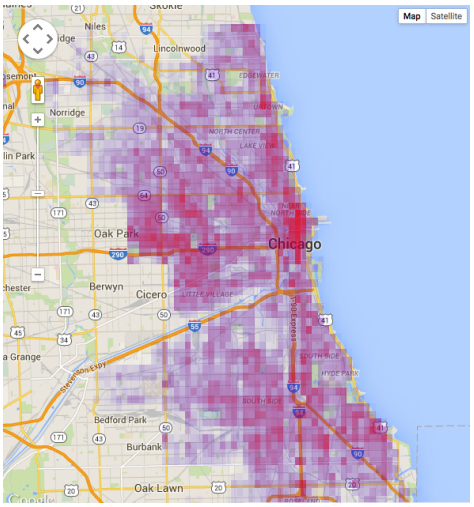


(a) Directions to 5000 N Pulaski Rd

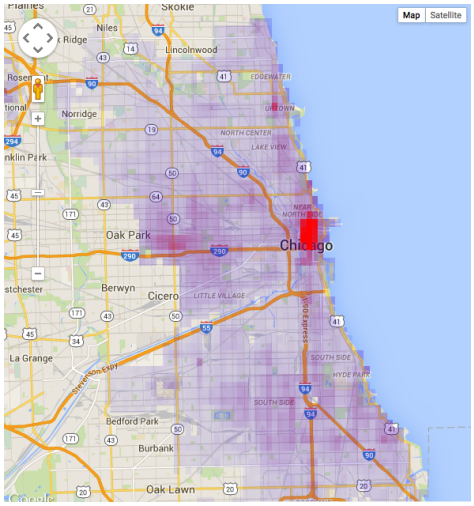


(b) Directions to 300 Pulaski Rd

Figure 5: High sensitivity to route choice.



(a) Probability that any crime will occur



(b) Prediction of how many crimes will occur

Figure 6: Comparison of 'any crime' probability score vs. 'how many crimes' score

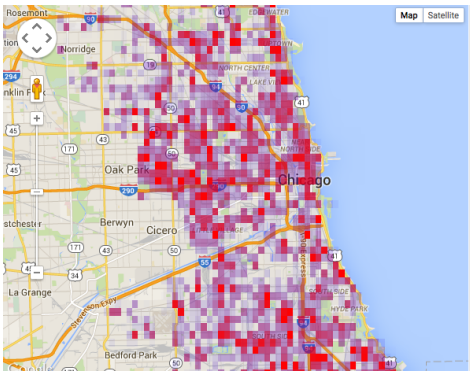
diction of **how many** crimes there are on a block. This would allow me to normalize by ridership to . In Figure 6, I compare a random forest classifier with a random forest regressor. The regression model clearly resembles the data more closely as one would expect.

3.5 Comparing hours

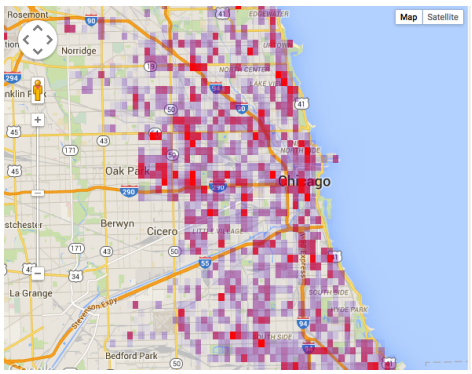
Looking in the month of July it was interesting to see how the general crime prediction varies during the course of the day. In Figure 7and Figure 8, we can see that the crime prediction is lowest from 4-8 a.m. and higher later in the day.

3.6 The Loop is your biggest enemy

Even when normalizing based on ridership, the downtown region still outweighs most places on probability of crime. That means that most routes that require any significant walking downtown will be a bigger cause for concern than most of the outlying dangerous areas. It's possible that this

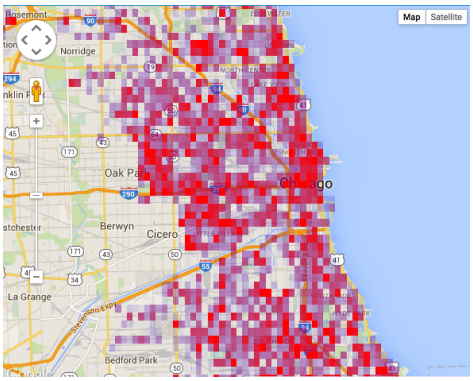


(a) Morning, 12 am - 4 am

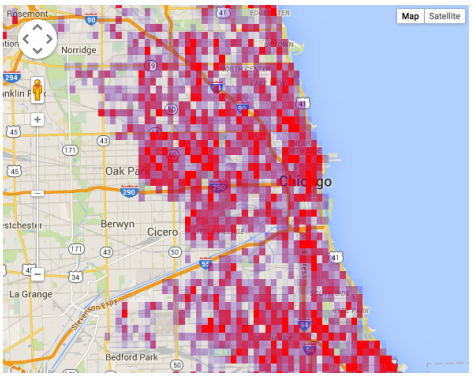


(b) Morning, 4 am - 8 am

Figure 7: Comparison of morning crime.



(a) Afternoon, 12 pm - 4 pm



(b) Afternoon, 4 pm - 8 pm

Figure 8: Comparison of afternoon crime.

is an overestimate, and it warrants further study, especially with regard to the possibility that ridership normalization isn't the best way to account for different foot traffic.

4 Future Goals

4.1 Display route stats

I'm currently just displaying the route with a color code based on its score. I would like to display an additional bit of information explaining the justification for the score.

4.2 Uber comparison

I've worked with the Uber API in python, but I need to incorporate the comparison into the visualization using javascript. This will come online soon.

4.3 Dataset expansion

I'm currently evaluating my prediction accuracy at using some days to predict other days of the same year. Ultimately I'd prefer to combine data from two years and see how well the first year can predict the second. This would help determine how well crimes could be predicted across years, which would be important.

4.4 Integration with Goldstein's result

The Goldstein lab is planning to release their results by the end of the summer. I hope to generalize my web interface so that it can use their prediction models, which will incorporate weather and demographics as well.

5 Work Summary and Bibliography

This section summarizes the work I did on this project.

I read the following articles, though I would not say I thoroughly read more than half of them (crime stat documents seem to be very long-winded

Table 1: My caption

Task	**Time**	**Explanation**
learning software	25%	Working with googlemaps and scikitlearn
gathering data	10%	polling route info and combining with crimes
data cleaning	20%	removing crime types, converting position info
data analysis	15%	trying out the various classification schemes
data presentation	30%	working to import everything into javascript

and "governmental"): [http://www.justice.gc.ca/eng/rp-pr/csj-sjc/jsp-sjp/rr02_7/rr02_7.pdf] [http://www.popcenter.org/library/crimeprevention/volume_13/03-groff.pdf][http://www.rand.org/content/dam/rand/pubs/research_reports/RR200/RR233/RAND_RR233.pdf [http://www.icjia.state.il.us/public/pdf/ResearchReports/CLEAR2004.pdf][http://www.palgrave-journals.com/sj/journal/v21/n1/pdf/8350066a.pdf]

New software: Google Maps (python and javascript), scikitlearn (python)

Data size: crime=55 MB, routes=2MB, ridership=1MB

Code lines: HTML: 100, javascript: 400, python: 400