Machine Learning for Crime Prediction

Rocio Perez
pecanoro@umich.edu

Matt McAllister
mattmcal@umich.edu

Sam Bean
sabean@umich.edu

ABSTRACT

The application of machine learning algorithms to crime prediction is an area that has not received the most attention. Many emergency responders have a short description of the situation but do not truly know what awaits them at the scene. The ability to use salient features of a crime to predict the probabilities of unknown obstacles would be invaluable for police officers across the country. In this paper we describe methods derived from applying machine learning to Chicago's annotated history of crime that prove we already have the tools to inform about a crime with only a limited set of features. We will also examine the effects and results of several different algorithms, problem formulations, and time windows.

1. INTRODUCTION

Crime prediction is a field that hasn't received substantial attention as far as the application of machine learning algorithms. Most work in the past revolves around density estimation and graphical estimations [1][3. Algorithms for this problem are unsupervised cluster or density estimation techniques. These may have an aesthetic appeal, but fall short when it comes to generalization and use for emergency responders. The large availability of up to date data suggests a supervised approach may have significant implications for feature extraction and prediction. The existence of such models would imply that given a short description of a crime, in the form of a dispatch call and relevant date/time/location information we would be able to inform on unknown attributes of the crime. The model would even be able to inform on whether a crime will be violent when the information being supplied to a police officer does not contain that fact. A brief attempt to integrate Twitter information will also be discussed in this supervised framework, but for the majority of this paper we will introduce multiple approaches for utilizing the large crime corpus maintained by the City of Chicago to develop multiple supervised classification tasks that suggest future work in supervised crime prediction has great promise.

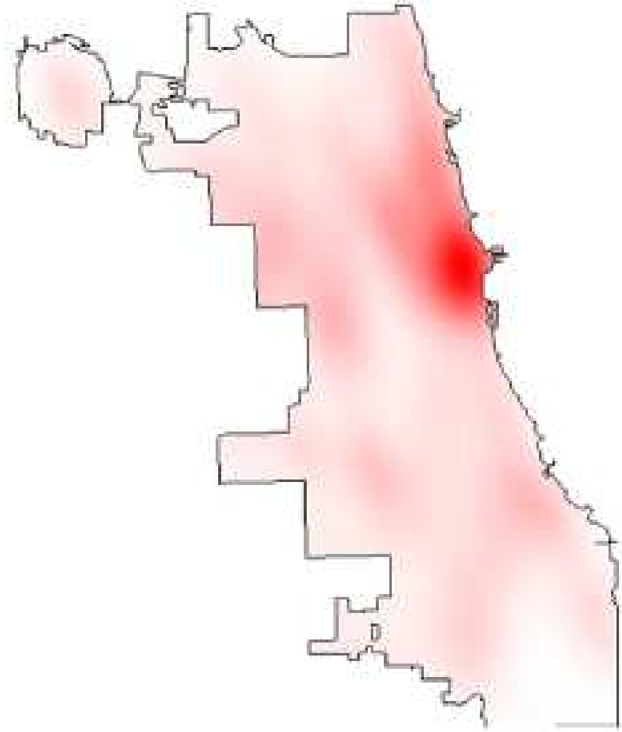


Figure 1: Heat map generated in Gerber's paper

2. RELATED WORK

2.1 Heat-Mapping

The most recent paper in crime prediction through the use of hot-spot maps was by Matthew Gerber during the end of last year[1]. Gerber used kernel density estimation to create heat-maps of Chicago using separate buckets (approximately 1000 meters by 1000 meters) and then smoothing the boundaries. The result for one crime is given above.

This work suffered in multiple ways: Gerber didn't use features of crime other than location to create the map, the maps were only generated on a per-crime basis which constrained the project significantly, and heat-maps are difficult to apply in a way that benefits police officers in the field. By that I mean any police officer will look at the generated maps and agree with them because they already have experience navigating the city and are well aware of the more dangerous aeas. Gerber then attempted to incorporate Twitter infor-

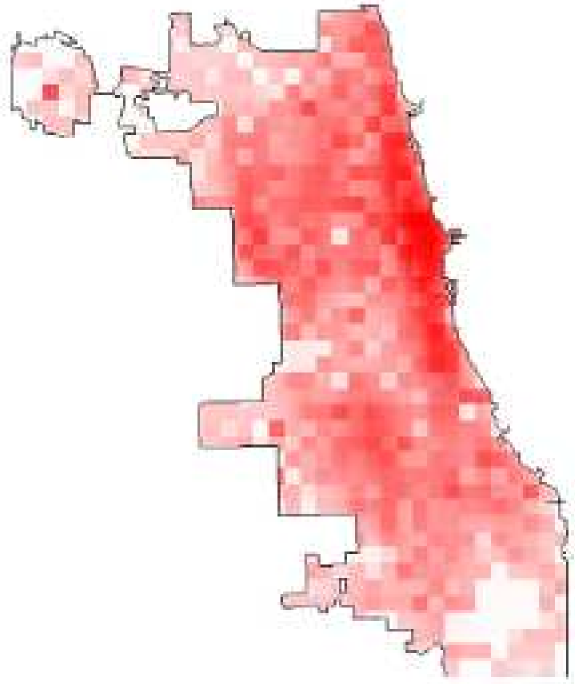


Figure 2: Heat map generated after incorporating Twitter features

mation based on if they originated from the given bucket. The resulting map, shown above, significantly darkened almost all areas. This corresponds to an increase of false positives given by the model. This implies that incorporating this model would introduce many logistic expenses and errors. Another paper on utilizing Twitter data is described next.

2.2 Twitter Event Extraction

Twitter is a data source that receives attention because of the amount and publicity of the data. In Wang's paper [2] he describes a method for using a local news station's Twitter feed to mine entries that have to do with crime. Semantic role labeling, notating text based on words that match named entities, is used to identify active portions of Tweets and whether there are important events in the text. From the events gathered, Latent Dirichlet allocation (LDA) is run to establish a distribution of topics that describe the text. This process iteratively moves through the given text and assembles topics, which are really collections of words, that best describe the collection. The topics are used in a predictive model that classifies new Tweets from the news station as either indicating a hit and run or not. The results were intriguing but struggled with high false positive rates and did not mention any results beyond the original training set.

2.3 Crime Prediction Techniques

Discovering crime patterns using previous data in order to prevent it has always been an important issue. However, the huge amount of data that must be analyze make impossible to people to examine every report manually, therefore, some techniques using computers has been developed during these years.

These techniques, examined by Grover et al[3], are classified into three different categories: statistical methods, based on age and offending behavior; geographical information systems that identify crime hot spots, crime attractors and crime generators; and machine learning techniques that involves patterns in criminal behavior.

However, these techniques has been prove not to be successful since most examinations were oriented to a type of crime or same offender. Grover et al suggest that in order to effectively predicts crime we have to focus on big geographical areas and consider all crime types.

In the same context, Wang et al[4] define and propose a crime detection algorithm called Serie Finder, a tool that learns from previous data to predict crime. This algorithm is based on two different coefficient which are capable of capturing of similar characteristics in crime. However, they also comment the difficulty of processing immense amount of data and how it is only possible to get good results when methods try to detect crime patterns by the same person or group, predicting crime that even analysts have missed.

2.4 Forecasting Criminal Behavior

On the other hand, different procedures and strategies have been studied for approaching the issue of accurate crime prediction. Richard et al[5] discuss and evaluates different statistical methods comparing them with the modern machine learning procedures. The article emphasizes the importance of the complexity of the decision boundaries which affects the forecasting accuracy, so the differences between them are presented in the document.

Three different classifiers will be analyzed and studied: logistic regression, random forests and stochastic gradient boosting. Even though these algorithms works well, logistic regression lacks the capacity that the other two have, defining more than two classes. However, this difference won't be such an important matter if boundaries are simple and good designed.

2.5 Police Narrative Extraction

Another problem which may be observed is obtaining meaningful information from narrative reports such names or places, so the data can be efficiently analyze by crime investigators. Chau et al [6] propose a neural-network extractor that feasible extract these attributes from reports. The results of the experiment shows that this system has a good performance retrieving names or drugs but not for addresses or properties.

3. DATA COLLECTION

3. Data collection Chicago maintains a massive corpus of crimes that have occurred from 2001 to the present. The

city is a great subject for large scale experiments on crime as it has the third highest population in the United States (2.7 million) and second in total murders, assaults, vehicle thefts, and a few others [7]. Each entry has a header whose attributes are shown below:

-ID
-Case Number
-Date
-Block
-IUCR
-Primary Type
-Description
-Location Description
-Arrest,Domestic
-Beat,District
-Ward
-Community Area
-FBI Code
-X Coordinate
-Y Coordinate
-Year
-Updated On
-Latitude
-Longitude
-Location

and an example of a crime is given here:

- 9524465
- HX179211
- 03/09/2014 10:30:00 PM
- 050XX W IOWA ST
- 910
- MOTOR VEHICLE THEFT
- AUTOMOBILE
- RESIDENTIAL YARD (FRONT/BACK)
- false
- false
- 1531
- 015
- 37
- 25
- 07
- 1142672
- 1905475
- 2014
- 03/14/2014 12:40:45 AM
- 41.89667358879907
- -87.75143977712743
- ”(41.89667358879907, -87.75143977712743)”

There was an almost unlimited number of training examples because of the large population and crime rate. We collected 3 training data sets of different sizes and 1 test data set. The sets are listed below with the respective time frame:

-10k: 3/05/2014 through 3/20/2014
-100k: 10/22/2013 through 3/20/2014
-250k: 5/07/2013 through 3/20/2014
-Test (7.5k): 3/21/2014 through 4/2/2014

We wanted the sets to overlap so there wouldn't be large discrepancies such as only having one season represented or missing holidays which have been linked to increases in crime in the past.

We also wanted to ensure the number of different crimes was not impairing or trivializing the predictions. Therefore, we also formulated the problem by lumping crimes into either violent or nonviolent crimes. Violent crimes are assault, battery, criminal sexual assault, and homicide as defined by the FBI Uniform Crime Report [7].

4. DATA PROCESSING

4.1 TF-IDF OVERVIEW

Given a set of documents and a query, an important task in information retrieval is being able to obtain the most relevant documents to a query. The first approach would be removing all documents where the terms of query don't appear, but we still have to decide how to rank the remaining files.

Another approach would be counting how many times the term appears inside a document (frequency of a word), but this will give priority to words that are repeated more frequently in a document such as ”the”, ”a”, etc. This doesn't take into account importance words that a query can contain, so what we need is another approach that combines both concepts.

TF-IDF is the solution to this problem. It combines the frequency of a word and the inverse document frequency, which calculates the frequency of a term taking into account all documents we are trying to process. Therefore, we can define tf-idf as:

$$TF - IDF = tf * log(\frac{N}{df})$$

Where tf is the frequency of a term inside a document, N is the total number of documents and df is the number of documents containing the term.

4.2 CRIME AS A DOCUMENT

Previous works about crime prediction have described how to use it-idf in order to obtain meaningful concepts from narrative police reports [6], but in our case, we are dealing with a file containing a list of attributes related to the crime, where each line belongs to a different crime.

In our project, we want to be able to process the data inside the document in order to use classifications methods that allow us to find out which attributes are the best ones to predict crime. Not all of them will have the same relevance so we need a measure that weight each one of the fields of our document.

The solution to this problem was creating a parser that use the tf-idf measure to transform each of the attribute fields to a numeric value. This parser will receive a target attribute that won't be modified and will become the class labels for our classifiers.

4.3 FINAL PROCESSING SCRIPT

We previously commented that in order to process our data using some classification methods, we needed to transform the attribute in the files to a more suitable form to successfully apply the algorithms, numeric values.

Our parser won't work the same way for each attribute because we don't have only string values, we also have boolean values or fields that are worthless for our classifiers like case number or ID. These fields will be transformed to an equal numeric value, so the information gain for these attributes will be inexistent and won't affect the results of the algorithms.

On the other hand, we have the string values, which are the most important values to take into account. At the beginning, we considered using a bigram representation in order to calculate id-tdf, however, we noticed that most of our words are used as key worlds, so using tf-idf with whole words returns the same result as using bigrams. Therefore, for each word in our document we calculate frequencies and using this information, we get a final result taking into account the total number of words for each field.

Finally, boolean values were easy to parsed since we only change the string representation for 0's or 1's. Numeric values remained the same since in our case, it doesn't make sense to calculate it-idf for these values.

The parser will return another .csv file with the same structure as the original one, but each one of the fields will be changed to their representation.

5. ALGORITHM DESCRIPTIONS

5.1 Support Vector Machines

Support Vector Machines (SVM) is a classification method that analyzes data and looks for patterns in order to a build a model which classify the data into a binary category.

This method assumes that data can be represented in a metric space. The classifier will search for a linear separation between classes. This linear classifier, called margin, will try to maximize the distance between data, where the closest samples to the margin will be known as support vectors.

5.2 Decision Trees

Decision trees can be used as a classification method to predict the value of a target variable given a set of input features. The decision tree will map each observation using feature names as nodes and feature labels as edges. Our leaves will be the values of our target variable.

Once we build our model tree, our testing samples will be predicted by going over the tree using our input feature labels until we reach a leaf, this leaf will be the label for our testing sample.

This is a snippet from part one decision tree used for predicting in the binary setting:

```
Location Description <= 20126
| Arrest <= 41747: non-violent (18858.0/345.0)
| Arrest > 41747
| | Domestic <= 41747: non-violent (10533.0/81.0)
| | Domestic > 41747
| | | Location Description <= 1695
| | | | Location Description <= 17
| | | | | Updated On <= 137760: violent (21.0/1.0)
| | | | | Updated On > 137760
| | | | | | Community Area <= 6282: non-violent (10.0/1.0)
| | | | | | Community Area > 6282: violent (2.0/0.0)
| | | | Location Description > 17: non-violent (4297.52/207.0)
```

5.3 NAIVE BAYES

Naive Bayes is a probabilistic classification method based on the Baye's theorem and uses assumptions of independence between variables.

In order to calculate the probability of a sample, we uses prior probability of each feature given no information about them. Since variables are independent, the model is defined by the joint probability of each feature:

$$\begin{aligned} p(C, F_1, \ldots, F_n) &= p(C)\ p(F_1, \ldots, F_n|C) \\ &= p(C)\ p(F_1|C)\ p(F_2, \ldots, F_n|C, F_1) \\ &= p(C)\ p(F_1|C)\ p(F_2|C, F_1)\ p(F_3, \ldots, F_n|C, F_1, F_2) \\ &= p(C)\ p(F_1|C)\ p(F_2|C, F_1)\ p(F_3|C, F_1, F_2)\ p(F_4, \ldots, F_n|C, F_1, F_2, F_3) \\ &= p(C)\ p(F_1|C)\ p(F_2|C, F_1)\ \ldots p(F_n|C, F_1, F_2, F_3, \ldots, F_{n-1}) \end{aligned}$$

P(C) is the prior probability of a class label and F1 to Fn are the features for our problem. This will create a posterior probability distribution that allows us to predict the outcome label.

5.4 LOGISTIC REGRESSION

Logistic Regression is a probabilistic classification method used to predict a binary dependent descriptor given one or more discrete or continuous independent features. The goal of this classification method is to estimate the probability of an event by fitting data to a logistic curve.

Logistic regression is a transformation of linear regression, however the function draws a s-shaped distribution, a logistic curve, which is more manageable for most applications.

$$F(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}}$$

Therefore, in logistic regression we will have a variable y which is the predictor label, usually binary, and a set of features that will be noted as x. With there variables x we will build a function which will be used to predict the output of y.

$$P = \frac{e^{a+bX}}{1 + e^{a+bX}}$$

5.5 K NEAREST NEIGHBOR

K-Nearest Neighbors is a simple and lazy learning method used for classification. Lazy means that the training data is not used to build a model nor perform a training phase, instead, it makes all calculations during the running phase. K-NN is usually used as a binary predictor, but it can be also used when we have more class labels

KNN makes some assumptions about the data. The training data must be able to be represented inside a metric model, meaning that we can calculate the distance between any two samples. Moreover, we are given a number k which refers to the number of the closest examples we are going to retrieve.

Therefore, we have our training data and a query and what we are going to do is to calculate the distance between the query and the training samples. Once we have our distances, we are going to retrieve the k nearest samples to the query, Then, we count the resulting labels of the k nearest examples and we will assign to the query the label which has majority.

6. TWITTER DATA AND MODELING

The official police records had numerous fields that made predicting crime type possible, but not entirely accurate. Our next step was to integrate data from Twitter into our models that were previously created using the official crime records in an attempt to increase the accuracy of detecting which type of crime is likely to occur at specific locations and times. Although the only relevant information that tweets contain is location, time, and the message text, we were optimistic that by integrating tweets into our models the accuracy of our system would improve. In order to include Twitter data in our system, we first collected as many tweets from users as possible. Next, we used topic modeling to predict if a tweet's message was referenced a crime. Finally, we added the tweets that were about crime to the set of police records.

Since the police records we had available were from the Chicago area, we chose to only focus on tweets that originated around Chicago. The figure above shows a bounding rectangle around the area where tweets were collected. Our original plan was to use Twitter's REST API to filter results not only by location, but also by keyword. However, despite the fact that we searched for tweets within a specific geolocation, the tweets that we received did not contain the exact latitude and longitude coordinates, but instead had a generic city name or a bounding polygon. Additionally, Twitter places a strict limit on the number of tweets that can be retrieved per day using the REST API, which results is a small dataset.

In order to get a larger dataset as well as a precise location of each tweet we used Twitter's streaming API. Unfortunately, the streaming API does not allow searches to be filtered by both location and keyword, so anytime someone in Chicago (with location services enabled) tweeted, we received it. Over the course of about two weeks we were able to collect over a million tweets. Since we were unable to filter tweets by content, most of them had nothing to do with crime.

Our next step was to determine if a tweet was about crime or not. In order to accomplish this we used the ARK processing tool [10] to tokenize and stem the text of each tweet. This natural language processing tool is specifically designed for Twitter as is correctly handles emoticons, hashtags, common abbreviations, as well as common misspellings. We then used MALLET [8], a collection of natural language processing tools, to perform LDA topic modeling on the tokenized tweets. The plate notation of LDA is shown on the facing page:

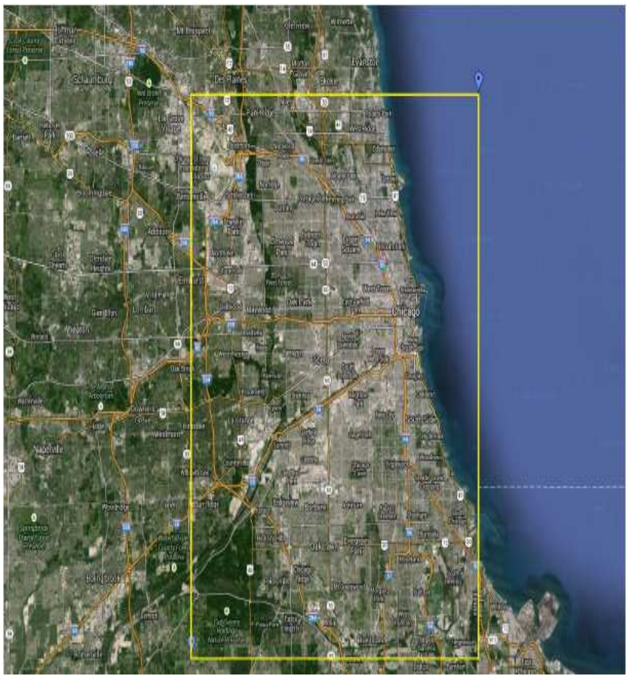


Figure 3: Boundary around Chicago Tweets were collected from

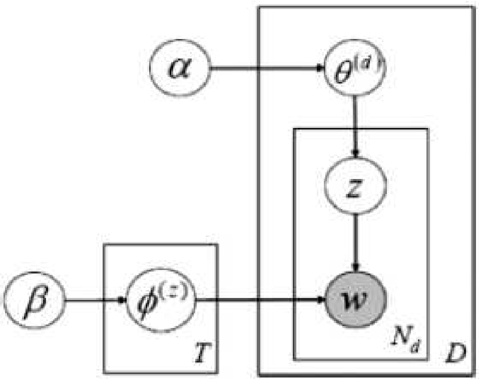


Figure 4: Plate notation for Latent Dirichlet allocation

Next, we assigned a topic id to the entire tweet based on the most common topic among its individual words. After analyzing the words that defined each topic we were able to decide which topics could be relevant to crime. Tweets with these topic ids were added to the list of crime tweets.

A second approach we used to identify tweets that were about crime was to search the tweets for terms that indicate crime, such as gang and drug dealer. The following is an example of a tweet that was found using this method: ”Drug dealers, gangbangers, and shooters wats on my block”. Tweets that contained at least one of these terms were also added to the list of crime tweets. Now we have a list of tweets that we have deemed to be about crime as well as a specific location and time when each was sent.

Unfortunately, this list of tweets about crime is very short, and after reviewing several of the tweets, it is clear that they are not all actually about crime. Determining if a tweet is about crime proved to be a very difficult challenge; since most of the tweets that we receive were not about crime, the topics that were identified were also not about crime. For our dataset topic modeling proved ineffective at identifying crime tweets. It is interesting to note that although the topics were not about crime, most of the topics contained words that were similar to each other. The list below displays the words that belong to each topic. Two particularly interesting topics were topic 0, which correctly grouped many curse words together, and topic 5 which contains all Spanish words. Had we been able to filter tweets by content as well as location it is possible that topic modeling would have been far more effective as it could have created topics that were divided by type of crime.

0 0.13391 shi* f*ck a*s bi*ch fu*king ain ni*ga twitter yo im break bro crazy ni*gas gotta man bi*ches lil spring

3 0.08496 phone big mad room family mind times anymore tired isn forever bc called af weird brother sister bored asl

4 0.08187 love today real baby pretty world follow song hear rain babe wouldn gym boy couple hope future couldn low

5 0.01966 de la el en te se mi es lo los por con si para al na le tu las

7 0.05742 http chicago il park job airport jobs city international hare club office ord cta center illinois north mi lake

8 0.05737 happy ricky friends birthday money live coming food party cold reason full wanted drinking beer drink america pi*sed episode

9 0.12843 lol girl miss talk haha tho lmao ll text funny yeah talking friend hell mom im omg ya gonna

13 0.06535 long person april thought stay fools hurt video mine dog joke bae number deal shower feelings scared strong instagram

16 0.08508 back life ll find music summer follow ur point sad happen drive boyfriend story listen ride snapchat dream means

17 0.07405 don wanna care car leave heart movie kids dead understand sos walk college drunk forget problem suck bye trip

18 0.08424 tonight gt bed girls perfect amazing thinking buy picture boys meet free line wear read open excited blue month

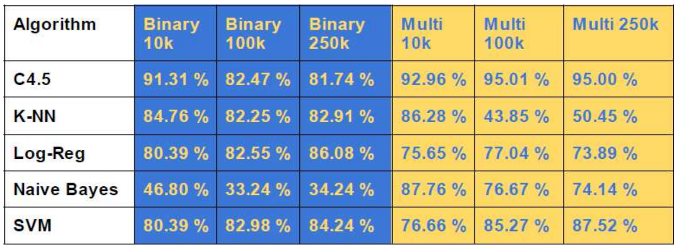
19 0.07092 night sleep stop hours hot late everyday fine nap water words chance playing mother gave met rock telling supposed

A second problem with the Twitter data is that only tweets that contained location data were collected, and only about one percent of tweets actually contain a geo-location. This resulted in a large amount of data not being collected. The final and most significant issue with integrating Twitter into our system is that no new information was gained. Data obtained directly from the official criminal records was superior to using data from Twitter, so it is not included in our final system.

7. EVALUATION AND RESULTS

The data was fed into Weka [9] for several machine learning experiments. We had the 3 data set sizes which were split into both the binary and multi class formulations.

Using the five algorithms described previously we obtained results on all 30 runs which are given below.



Algorithm	Binary 10k	Binary 100k	Binary 250k	Multi 10k	Multi 100k	Multi 250k
C4.5	91.31 %	82.47 %	81.74 %	92.96 %	95.01 %	95.00 %
K-NN	84.76 %	82.25 %	82.91 %	86.28 %	43.85 %	50.45 %
Log-Reg	80.39 %	82.55 %	86.08 %	75.65 %	77.04 %	73.89 %
Naive Bayes	46.80 %	33.24 %	34.24 %	87.76 %	76.67 %	74.14 %
SVM	80.39 %	82.98 %	84.24 %	76.66 %	85.27 %	87.52 %

Figure 5: Results across problem formulation, data set size, and algorithm

It seemed support vector machines and decision trees were the best of the candidates for several reasons. Decision trees obviously had the best performance but many splits were based on a small subset of the features which pointed to over fitting. Support vector machines scaled well with the training data and performed well on both formulations.

We used these algorithms to test against the future set but suspected the results may be poor from over fitting. The decision tree algorithm supported this hypothesis as accuracy dropped steeply. However, support vector machines only suffered a 5% decrease in accuracy when trained with the largest data set. This result is extremely exciting as it points to strong trends in a corpus with a limited number of features. The existence of a discriminative algorithm points to a corresponding generative model.

8. FUTURE WORK

The future for supervised crime prediction looks bright indeed. A generative network such as a deep belief network could be formulated and actualize a real time crime analysis engine. By that I mean certain features, such as date, type of crime, time, or location, could be ”clamped” and the model would generate missing features with a high probability. This engine could take a description in the form of a transcribed dispatch call, a location, and other salient features. It would then give officers the information missing by matching against the growing corpus of crime. With the limited resources and time frame this group faces, we were not able to scratch the surface of the information available. We hope work in the future is able to process and model every crime from 2001 to present. Generative models that resulted from such a vast data supply could save innumerable lives for emergency responders across the country.

9. PROJECT MEMBERS

Rocio Perez dealt with pre-processing the data and building the logistic regression, k nearest neighbor, and naive bayes models.

Matt McAllister streamed data from Twitter, processed the corpus, and built topic models.

Sam Bean pre-processed data, devised the training schemes, and tuned the SVM and decision tree models.

10. REFERENCES

[1] Gerber, Matthew S. ”Predicting crime using Twitter and kernel density estimation.” Decision Support Systems 61 (2014): 115-125.

[2] Wang, Xiaofeng, Matthew S. Gerber, and Donald E. Brown. ”Automatic crime prediction using events extracted from twitter posts.” Social Computing, Behavioral-Cultural Modeling and Prediction. Springer Berlin Heidelberg, 2012. 231-238.

[3] Grover, Vikas, Richard Adderley, and Max Bramer. ”Review of current crime prediction techniques.” Applications and Innovations in Intelligent Systems XIV. Springer London, 2007. 233-237.

[4] Wang, Tong, et al. ”Learning to detect patterns of crime.” Machine Learning and Knowledge Discovery in Databases. Springer Berlin Heidelberg, 2013. 515-530.

[5] Berk, Richard A., and Justin Bleich. ”Statistical Procedures for Forecasting Criminal Behavior.” Criminology and Public Policy 12.3 (2013): 513-544.

[6] Chau, Michael, Jennifer J. Xu, and Hsinchun Chen. ”Extracting meaningful entities from police narrative reports.” Proceedings of the 2002 annual national conference on Digital government research. Digital Government Society of North America, 2002.

[7] United States Department of Justice, Federal Bureau of Investigation. (September 2012). Crime in the United States, 2011. Retrieved 4/2/2014.

[8] McCallum, Andrew Kachites. ”MALLET: A Machine Learning for Language Toolkit. http://mallet.cs.umass.edu. 2002.

[9] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009); The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1.

[10] Improved Part-of-Speech Tagging for Online Conversational Text with Word Clusters Olutobi Owoputi, Brendan O’Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider and Noah A. Smith. In Proceedings of NAACL 2013.