RoboCop: Crime Classification and Prediction in San Francisco

John Cherian, Mitchell Dawson

Purpose & Summary

Resource allocation is not just a serious problem for multinational corporations and other large businesses. Knowing how to allocate resources is a pertinent issue for police departments across the United States. Having the right personnel on the street requires knowing what type of crime is likely to occur at a specific time and location. The goal of this project, therefore, is to classify crimes based on location and time metadata as well as predict surges and falls in specific categories of crime before they happen.

To accomplish this goal, we applied supervised machine learning techniques, and in particular a random forest model, to classify crimes, ultimately achieving an out-of-sample test set error rate of 68.3% and a training error of 25.3%.

We also fit a time series model which tracked the number of instances of each class of crime over a biweekly period. This model, though imperfect, delivered promising results, with a root-mean-squared error average of approximately 63 on an out-of-sample test set.

Dataset

San Francisco Crime Classification is a Kaggle competition to determine whether one can effectively predict the category of crime that occurred based on metadata about the crime. This metadata includes the following features: Date, Day of Week, Police Precinct, Address, Latitude, and Longitude, i.e. time and place.

The Kaggle dataset covers every crime report made between 1/1/2003 and 5/10/2015. However, only every other week has labels (the other weeks are set aside as a test set).

Because of poor performance on the Kaggle dataset, we attempted to supplement the crime metadata with data about the neighborhood in which the crime took place (data from Census' American Community Survey), seeing if changing demographics or incomes in neighborhoods might also help predict the types of crime that might occur, e.g. increasing wealth in the Mission increases the likelihood of property crime when compared to violent crime.

Preliminary Work

Algorithm	Training Set Error	Test Set Error
Logistic Regression	80.00%	80.01%
SVM	NA	NA
Naïve Bayes	77.26%	77.27%
Random Forest	15.32%	70.69%

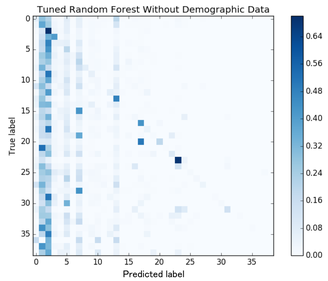
Our first step in classifying crimes based on metadata in San Francisco was to try various off-the-shelf classification models.

The large difference between training and test set error in our preliminary result for random forest suggested that we were suffering from overfitting. However, the large size of our dataset and the computational burden of recalculating each model made using conventional feature selection techniques impossible.

The other models were little better than the naïve classifier that always chooses the most common label. The SVM also took too long to run on any reasonably sized training set.

Tuning Random Forest

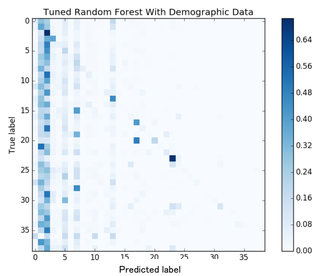
Because of the relative effectiveness of random forests, we first sought to fine-tune the parameters of this model. The two parameters important for our purposes are the number of estimators (the number of decision trees created by the model) and the maximum depth of each tree in the forest. Our preliminary tests were performed with no maximum depth and a small number of estimators. The lack of maximum depth in particular likely contributed to the overfitting we observed.

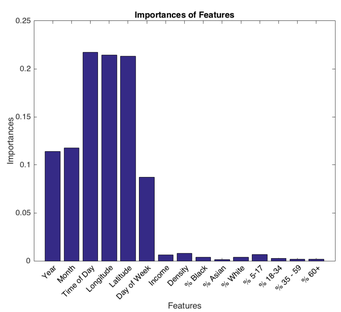


Due to the size of our dataset, we were constrained in our ability to experiment with the number of estimators, as the time required to fit the model became prohibitively long. We found that increasing the number of estimators strictly improved the performance of the model in terms of both training and test errors, and we had not hit diminishing returns in this respect by the time we reached a prohibitively large number. To the left is the confusion matrix generated by a tuned random forest.

Incorporating Demographic Data

Using data from the 2005-2009 American Community Survey and the 2010 Census, we created demographic profiles for each of the police precincts in our data set. These profiles included data on per capita income, housing density, racial composition, and age. We used these statistics to create new features. The confusion matrix to the right was generated by a random forest with the same parameters as above run on this new, expanded dataset. The difference between the two matrices is negligible.

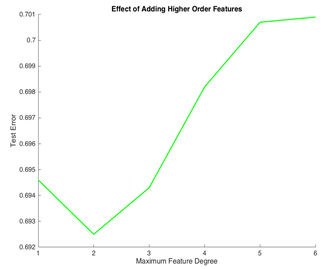




We achieved a test set error rate of 68.94% when including the demographic data, which is a hair higher than our error without such data. To the left is a chart of the importances of the various features in the random forest. It shows that the added demographic features were uninformative in the classification task. For this reason, we did not pursue any further experimentation with the demographic data.

Adding Features

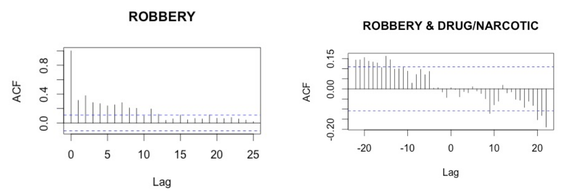
We next tried to reduce the bias of our model by building higher order features (such as longitude squared or time of day times latitude) and adding them to our feature set. Because of the large size of our data set and consequent long training times with our initial set of features, we chose to only work with the 3 most important features: time of day, latitude, and longitude. For a given degree n, we calculated all higher order features of degree less than or equal to n involving those three features and added them to our feature set. The plot below shows how the test error rates changed as we varied n.



Because we only wished to see how the error changed with the degree, we ran these tests on random forests with a more manageable number of estimators than before. We found that there was a very slight improvement in test error with the addition of quadratic features, but performance suffered with n > 2.

Time Series Analysis: Model Selection

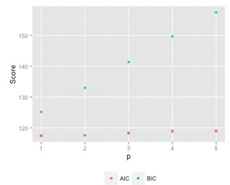
While classifying crimes based on metadata may be relevant to San Francisco city officials, a more interesting and perhaps broadly applicable problem focuses on predicting crime before it happens. The motivation for time series analysis being a valid approach for this problem is as follows. Auto-correlation function (ACF) and cross-correlation (CCF) plots between the different categories of crimes revealed significant statistical connections that a multivariate time series model can capture. The following are two particularly informative plots.



To capture the cross-correlation between different time series, we employed the VAR(p) (Vector Auto-Regression) model defined as follows:

$$N_t = \omega + \sum_{j=1}^{p} A_j N_{t-j}$$

Choosing p for this model is a feature selection problem. However, in this case, instead of using cross-validation, we fit the model with different values of p, and calculated the model AIC (Akaike Information Criterion) and BIC (Bayesian Information Criterion). As you can see, p = 1 appears to be optimal.

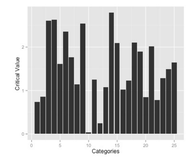


Time Series Analysis: Fit and Results

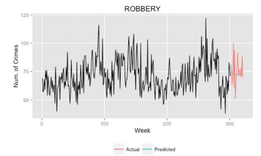
The VAR(p) model has an important assumption of stationarity, i.e. the time series satisfies the following conditions:

$$E[N_t] = \mu \quad \forall t \qquad Cov(N_t, N_{t-j}) = \gamma_j \quad \forall j$$

We test for stationarity of the model using the Augmented Dickey-Fuller Test. The critical value observed for each categorical time series is significant at the 5% level if it exceeds 2.86. Though some do come close, none of the critical values exceed that threshold.



After fitting the model using Yule-Walker (Method of Moments), the graph to the right documents how the model performs on a prediction that goes 20 weeks out into the future. One of the properties of the model is that the accuracy of the prediction decays over time (as it is further removed from real data) and simply reverts to the mean. This is clearly observed in the graph to the right.



To evaluate model error, we cross-validated the model fit on a moving window of size 100. The categories with the largest normalized RMSE tended to exhibit some sort of linear trend too insignificant to cause a rejection by the Dickey-Fuller test, but substantial enough to affect the usefulness of our predictions. In our paper, we present a detrended time series model as well to control for these changes.

Category	Normalized RMSE	Larceny	25.587	Stolen Property	10.946
Assault	2.384	Missing Person	9.374	Suspicious Occurrence	14.407
Burglary	10.788	Non-Criminal	29.944	Trespassing	2.372
Disorderly Conduct	6.229	Other Offenses	14.693	Vandalism	7.536
DUI	6.857	Prostitution	12.908	Vehicle Theft	68.599
Drug	28.566	Recovered Vehicle	14.783	Warrants	8.204
Drunk	2.855	Robbery	7.607	Weapon Laws	3.225
Forgery	23.466	Secondary Codes	11.35		
Fraud	2.525	Sex Offenses (Forcible)	5.090		
Kidnapping	4.580				

Future Steps

First, with additional time, adding data from cities like Chicago would help train more accurate models. However, our location data would have to become a relative metric such as distance from city center, formats would need to be aligned, etc.

For time series, using the MINGARCH(p, q) model that John developed in his summer research would be useful. Unlike the Gaussian assumption in VAR(p), INGARCH assumes the counts are drawn from a Poisson distribution (much more appropriate for crime statistics). Unfortunately, the currently implemented version of the model is insufficiently optimized for a larger dataset like this one.

References

A. Nasridinov, S. Ihm, and Y. Park. A Decision Tree-Based Classification Model for Crime Prediction. *Information Technology Convergence*, 531-538, 2013. PDF.

K. Kianmehr, R. Alhajj. Effectiveness Of Support Vector Machine For Crime Hot-Spots Prediction. *Applied Artificial Intelligence*, 433-458, 2008. PDF.

A. Bogomolov, et. al. Once Upon a Crime: Towards Crime Prediction from Demographics and Mobile Data. *MIT*, 2014. PDF.

T. Wang, C. Rudin, D. Wagner, and R. Sevieri. Learning to Detect Patterns of Crime. *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, 2013. PDF.

All data is from Kaggle or the U.S. Census