

# A Crime Forecasting Tool for the Web-Based Crime Analysis Toolkit

Mark B. Mitchell, Jr., Donald E. Brown, James H. Conklin

**Abstract—** In this paper, we outline the development of a crime forecasting tool for law enforcement applications. This tool implements a spatial Discrete Choice Model (DCM) that takes into account the attributes of locations and the preferences of criminals in selecting target locations. Other steps required for development were the acquisition of data and the selection of suitable software components. Once these design decisions were made the components were integrated with the Web-Based Crime Analysis Toolkit to make the tool available to crime analysts throughout Virginia.

## I. INTRODUCTION

CURRENTLY, a vast majority of law enforcement agencies do not use tools for forecasting the future locations of crime [2]. As a result, the goal of this project was to build a crime analysis tool implementing a spatial prediction algorithm. To increase the availability of the tool, a secondary goal was to integrate the tool with the WebCAT (Web-based Crime Analysis Toolkit) system, which is currently being deployed across Virginia [1].

To develop the tool, a number of decisions were made affecting the accuracy, speed, and overall performance of the forecasting tool:

1. Choice of Algorithm
2. Software technologies
3. Data

In this paper, we describe the development of the tool for making forecasts in Richmond City, and the necessary steps for expanding capability to forecast crime in other Virginia jurisdictions. Furthermore, this paper will show prediction results for several crime data sets.

## II. CHOICE OF ALGORITHM

Central to the functionality of this tool is the choice of the algorithm (process) by which it makes its predictions. Three categories of algorithms were considered for this application: hot spot methods, spatial discrete choice models (DCMs) using linear prediction functions, and DCMs using nonlinear prediction functions.

1. Hot spot methods: Hot spot methods are a collection of techniques that identify clusters of points [5], which are labeled as hot spots. In some

methods these hot spots are defined and identified using a formal statistical definition, and in some methods these hot spots are defined and identified using less formal techniques. Once the hot spots have been identified, the interpretation is that crime has been dense in these regions and will continue to occur in these areas [5].

2. Spatial discrete choice model (DCM): DCMs model the criminal decision process to predict the location of future incidents [5]. The discrete choice model predicts that incidents will occur in areas similar in feature space to the areas where they have occurred historically. Within the class of discrete choice models, there are several techniques for performing the prediction. Both linear and nonlinear models have been used, and each of these was considered for use with WebCAT.
  - a. Spatial discrete choice models using a general linear model (GLM) make their predictions based on a linear, weighted combination of the predictor features. This method runs quickly and is easily interpreted.
  - b. Spatial discrete choice models using a general additive model (GAM) make their predictions based on a nonlinear combination of the predictor features. This method is more accurate than the GLM, but is more computationally complex.

In determining which algorithm was better suited for this application, it was necessary to find an acceptable tradeoff between the time required to execute the forecast and the accuracy of the predictions

Hot spot methods have an advantage when it comes to the time required to generate a forecast. Hot spot techniques are faster because they are much less computationally complex than spatial discrete choice models. Another advantage of hot spot methods is that their outputs are easier to interpret: the concept of a cluster of points is intuitive.

The disadvantage of hotspots, as found by Brown and Liu, is that they are less accurate than spatial choice models [4]. The primary reason for this is that hotspot methods fail to account for the attributes of the incident locations, which are what trigger the choices made by criminals in selecting a target site. As conditions almost inevitably change, the preferences of criminals will also change. Since hotspot methods do no account for these preferences, this in turn means that the predictions from hotspot models will become invalid [5].

---

Manuscript received April 9, 2007. This work was supported in part by the Virginia Department of Criminal Justice Services (VDCJS).

Mark B. Mitchell is with the University of Virginia, Charlottesville, VA 22904 USA (e-mail: mbm8g@virginia.edu).

Donald E. Brown is with the University of Virginia, Charlottesville, VA 22904 USA (e-mail: deb@virginia.edu).

James H. Conklin is with the University of Virginia, Charlottesville, VA 22904 USA (e-mail: conklin@virginia.edu).

As mentioned previously, it was necessary to find a tradeoff between speed and accuracy to select an implementation method. We initially chose to use a linear DCM since previous comparisons have found these methods are more accurate than hotspot methods [4], yet they run faster than nonlinear DCMs. We later compared the linear DCM to a nonlinear version of the DCM to verify we made the correct decision. In order to evaluate the speed and accuracy of these techniques, test scripts were used along with a sample data set to produce forecasts. In producing these forecasts, we were able to determine the speed of each method. Since the sample data set comes from past data, we were then also able to compare the forecasts to what actually happened as a measurement of the accuracy of the forecasts. The details of this comparison are described in the testing section.

### III. SOFTWARE TECHNOLOGIES

In executing either of the algorithms discussed above, it is necessary to make use of specialized software packages to carry out computations. The necessary computations can be divided into two categories: statistical and geographical.

One consideration for the statistical package was the necessity for the chosen package to be able to handle large amounts of data while performing statistical calculations. The exact nature of the calculations depends on the algorithm being used. Additionally, the cost of each software package in terms of both money and time for implementation were considered.

The available options for a statistical were the open source package R, custom-written code to perform the calculations, or the statistical package MATLAB. The time for implementation was approximated by conducting research and applying our previous knowledge and experience. Our analysis is summarized in Table 1 below.

	Monetary Cost	Integrability	Data Handling	Development Time
R	5	4	4	4
Custom Code	5	3	4	1
Matlab	3	3	4	3

Table 1: Evaluation of Statistical programs

In the table, each option considered was given a 1-5 rating in each category, with 1 being the worst score possible and 5 being the best score possible. The ratings from each of these categories were then added (each category was weighted equally) and the highest scoring option was selected.

The other category of software considered was Geographic Information Systems (GIS). A GIS is a program that allows users to create maps and link elements of the map with data stored in a database. Additionally, GIS provide capability for performing spatial operations. The spatial analysis capabilities of a GIS will be necessary in preparing the data for making forecasts, such as computing distance features.

Similar to the statistical packages, the GIS perform necessary calculations while handling large amounts of data.

The programs considered include ArcGIS, Manifold, and GRASS. To evaluate the suitability of each option, the speed, ease of integration, and monetary cost were considered.

The time for implementation was approximated by conducting research and applying our previous knowledge and experience. Additionally, we consulted the documentation for each product to gauge the degree to which we can integrate the packages with WebCAT. The results of our statistical package analysis are summarized in Table 2 below using the same evaluation criteria used for selecting the statistical package.

	Monetary Cost	Integrability	Data Handling	Development Time
Manifold	4	4	4	4
ArcGIS	1	4	4	4
GRASS	5	2	3	3

Table 2: Evaluation of GIS

### IV. PREDICTION GENERATION

#### A. Data

A critical component of the system is the data used to generate the forecasts. There are two categories of data used to generate the forecasts: feature data containing attributes of locations throughout the area of interest, and incident data containing the locations of previous incidents.

There are two types of features included in the feature data: demographic features and distance features. Demographic features include population details such as households per square mile and per capita income. The distance features measure the distance to specific landmarks such as interstates or highways.

The incident data we used contain the latitude and longitude for assault and burglary incidents from May through July in 2006. The data were grouped by type of incident, and within each incident type were grouped by month. This grouping resulted in a set of six sample data sets that contained either assault or burglary incidents for a specific month. Within each of these data sets, incidents from the first three weeks of that month were used as a training set to generate forecasts. For evaluation, the incidents from the fourth week of each month were used as a test set for comparison to the forecasts generated using that month's training set.

#### B. Execution

The forecasts are generated by considering a uniform grid of points that cover all of Richmond City. Each of these points has the demographic and distance attributes discussed previously. The tool compares each of these grid points to the incident points that are being used as the training set. If a grid point is the closest point to one of these incident points, the grid point has an indicator variable that is set to one, otherwise the indicator variable defaults to zero. Once all incident points have been paired with a grid point, the tool fits a model to predict the indicator variable value as a function of the grid point attributes. The model used is given by Equation 1 [3]:

## V. TESTING

$$\pi_i(x) = \frac{\exp[B_0 + B_1x_{i1} + \dots + B_kx_{ik}]}{1 + \exp[B_0 + B_1x_{i1} + \dots + B_kx_{ik}]}$$

Equation 1: Equation used for DCM (Brown et al. [3])

In this equation,  $x_{ik}$  is the  $k^{\text{th}}$  attribute value (such as distance to interstate or households per square mile) of the  $i^{\text{th}}$  grid point.  $B_k$  represents the coefficient that was fit to the  $k^{\text{th}}$  attribute to explain the observed indicator variable values. Once these coefficients are obtained, the tool reevaluates the model, but replaces the values of  $x$  with the actual attribute values of the grid points and the fitted values of  $B$  to predict the probability of a future incident occurring at that grid point. The output from the model is a probability for each grid point that a future incident will occur.

Initially, all available features were used to fit the model. After this initial fit, the model was compared to other models that excluded certain terms. Reducing the model served to improve the parsimony of the model by reducing terms that were either insignificant or highly correlated, while minimizing a decrease in the accuracy of the fit.

The output from the algorithm is then converted into a graphical display called a threat surface. The threat surface uses shading or coloration to show the probability of a future crime occurring at every location throughout the area of interest. In order to aid in interpretation, a threat surface is generally overlaid on a map of the area of interest. To further aid in interpreting the results of a forecast, the map can include other data layers such as roads and buildings.

Figure 1 below shows an actual output from the current implementation of our tool, where green represents low probability of future incidents and red represents a high probability of future incidents. The forecast is for Richmond City and was generated using assault incidents from May 1-24, 2006. The white points show actual assault incidents from the following week, and provide an informal evaluation of the forecast accuracy.

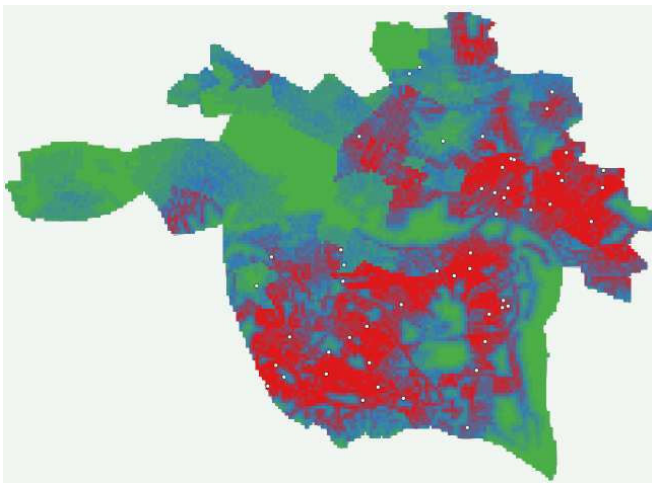


Figure 1: Threat Surface Prediction of Assault Incidents

We chose to use the linear DCM in the development of the tool described here. To evaluate the performance of this algorithm, we compared it to a nonlinear DCM and evaluated the accuracy of forecasts using two methods. In both methods, a set of six sample forecasts were generated using three weeks of incident data as a prior sample. These forecasts were then compared to the incidents from the week directly following as a measurement of the accuracy of the forecasts.

The first comparison method used is similar to the percentile score of the density estimates, as discussed by Brown and Liu. In this method, the output probabilities are examined at locations where incidents occurred in the week following the three week prior sample. The probabilities at these locations are compared between multiple models: whichever model has the highest probability at a point where a future incident actually occurred is more accurate than the other model. This method is used as a comparison between alternative models, but does not convey much information when used for a single model.

The second comparison method used is counts of the future incidents that appear in the highest risk areas of the threat surface. To do this, we looked at how many incidents appeared in the top 3% highest risk areas of the threat surface, and repeated this calculation for the top 6%, 9%, 12%, and 15% highest risk areas. A useful way to display these results is to construct a plot with the percentage of the threat surface covered (the threat level) on the x axis and the proportion of incidents within that threat level on the y axis. By plotting this curve for multiple models on the same graph, one is able to get a sense of the performance characteristics of each model. An important characteristic shown by this graph is the accuracy of the forecasts: the steeper the curve, the more accurate the forecasts.

Two graphs we generated using this method are shown in Figures 2 and 3.

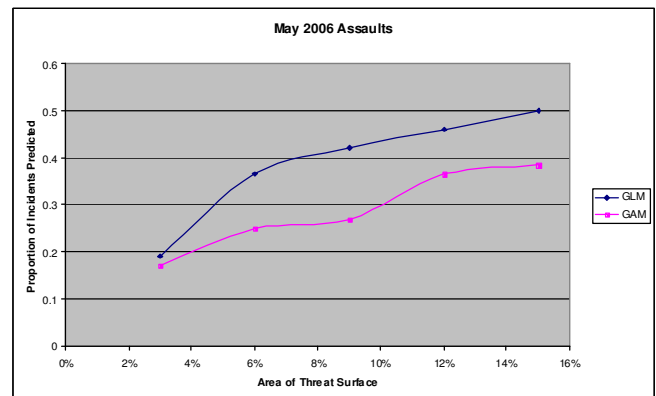


Figure 2: Percentage of Future Incidents Captured vs. Area of Threat Surface for May 2006 Assaults

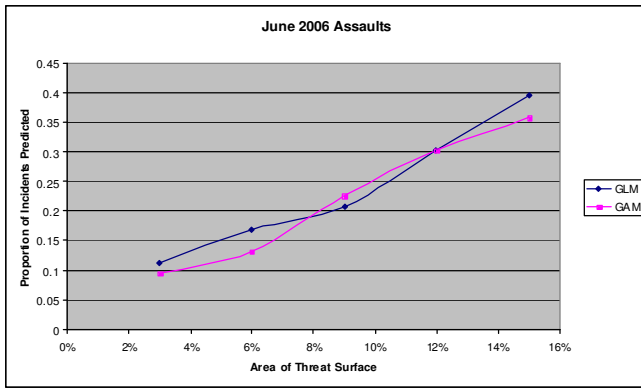


Figure 3: Percentage of Future Incidents Captured vs. Area of Threat Surface for June 2006 Assaults

In Figure 1, the GLM is more accurate, because at every threat level the GLM contains a higher proportion of actual future incidents. Although a GAM is usually more accurate, the GLM was more accurate in this scenario. The GLM is also more accurate in Figure 2. In this figure, the GAM is more accurate at the 9% threat level, both algorithms capture the same proportion of incidents at the 12% level, and the GLM is more accurate at the other threat levels. Since a GLM is also faster than a GAM, we retained it in our final design.

This metric is useful because it allows comparison between models, as with the previous metric, but also has significance for a single model. When used to compare more than one model, whichever model has the highest count of future incidents at most threat levels is the most accurate. When this metric is used to evaluate a model individually, the interpretation is less objective but it retains a useful measure of accuracy. For example, if all future incidents appear in the top 6% of the threat surface, this suggests the visual display allows a user to easily and correctly identify the highest risk areas. Since one goal of the tool is to allow law enforcement agencies to take proactive steps to fight crime, this is a desirable trait. On the other hand, if none of the future incidents appear in the top 9% of the threat surface, this suggests the forecasts are inaccurate and thus not useful to users.

## VI. INTEGRATION WITH THE WEBCAT SYSTEM

After selecting which algorithm and software packages to use, we integrated our tool into the WebCAT system. This section contains a description of the final design.

As with other WebCAT tools, users begin at the query selection page, shown in Figure 4. On this webpage, the user specifies parameters for the set of incidents that will be used to make the predictions. For example, the user might specify all assault incidents within the last three months if s/he wishes to anticipate what the patterns in assaults will be for the next month.

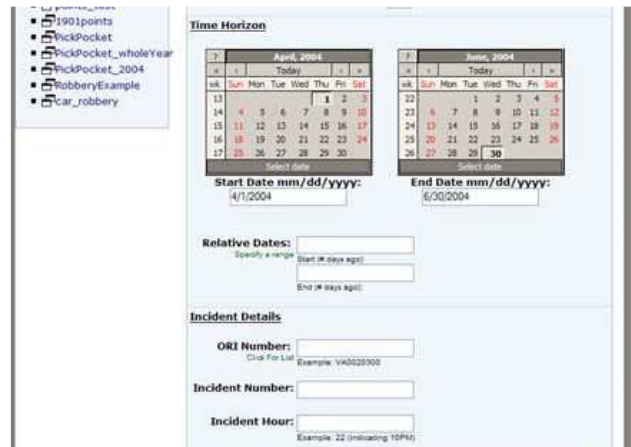


Figure 4: View of WebCAT Query Selection Page

Once the user specifies the desired query, WebCAT searches its database and returns all matching results, and then redirects to the query results page. At the query results page, the user is present with numerous links for the available analysis tools that can be used on the matching crime incidents, as shown in Figure 5. To use the forecasting tool, the user clicks on the link labeled "Forecasting".



Figure 5: View of Query Results Page

Doing this opens a new web page, which briefly redirects to a blank page while the tool runs. Once the tool has finished generating its forecast, the tool redirects to a web page displaying the threat surface generated. An example of this threat surface is shown in Figure 6.

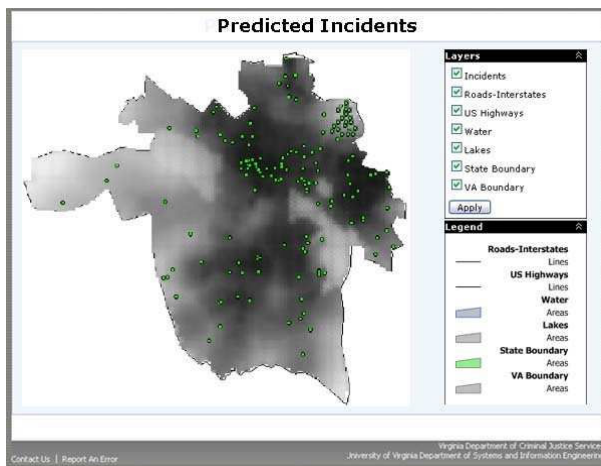


Figure 6: Representative Threat Surface, adapted from Brown and Liu [3].

## VII. APPLICATION TO LAW ENFORCEMENT

The law enforcement agencies in jurisdictions that employ the WebCAT system will be the group most directly influenced by this project. Jurisdictions will be able to more efficiently allocate and deploy their available resources. These changes in resource allocation could take several forms, such as a change in the beats of patrols. These reassignments would be made so that patrolling officers pass through areas where crimes are predicted to occur, during the time frame which they are predicted to occur. Consider breaking and entering as an example. Many of these events occur at residential locations, during the day, while no one is home. To account for this, beats could be changed to patrol more heavily in neighborhoods during working hours. At night, the beats could focus on patrolling commercial locations that are closed, when crime is more likely to occur. The goal of such reallocation would be to more efficiently catch criminals, and ultimately to lower crime rates as a result. To reiterate, the tool we developed will allow law enforcement agencies to improve enforcement without increasing resources, and may even allow them to intelligently divert resources from active patrols and still maintain the same level of effectiveness. The possibility of this situation is illustrated by a study undertaken by the Charlottesville Police Department to improve the beats of their officers.

The city of Charlottesville commissioned two University of Virginia researchers to analyze the patterns in the calls for service and recommend how to improve response times. The researchers examined the data and observed the trends in calls by time of day, day of week, and time of year. From this analysis, the researchers were able to predict call patterns, and so they recommended new beats for the officers. The new beats allowed officers to respond faster without increasing the number of officers on patrol [6]. Although this study was conducted by researchers from the University of Virginia, a predictive tool developed for WebCAT could allow users of the WebCAT system to conduct similar studies without needing to commission outside groups.

## VIII. CONCLUSION

This paper describes the development of a crime forecasting tool that integrates with the WebCAT system. The major steps in this project involved selection of a suitable algorithm, as well as appropriate statistical and GIS software packages. Once these choices were made, it was necessary to develop the different components of the system and then integrate them with WebCAT. Testing was conducted to verify the algorithm used performed acceptably relative to other algorithms, while also being fast enough to remain usable. This tool allows users of the WebCAT system to predict where crimes of a specific type are likely to occur in the future. This will allow law enforcement jurisdictions to allocate their resources more effectively, leading to a reduction in crime.

## ACKNOWLEDGMENTS

The authors wish to acknowledge Butch Johnstone of the Virginia Department of Criminal Justice Services for his support of the project, as well as providing suggestions for the project. The authors also wish to acknowledge Mike Lawton of DaPro Systems for providing insight into current crime analysis practices.

## REFERENCES

- [1] Almazinos, C., Bowman, D.C., Eagan, R.M., Kuklinski, T.R., Nguyen, D.D., Brown, D.E., Conklin, J.H., Hansen, P.J. WebCAT: The Development, Performance Analysis, And Deployment of a Web-based Crime Analysis Toolkit. (*Presentation at IEEE Systems and Information Engineering Design Symposium*, (2006).
- [2] Boba, Rachel. *Crime Analysis and Crime Mapping*, Sage Publications: Thousand Oaks, 2005.
- [3] Brown, D.E., Dalton, J., Hoyle, H. Spatial Forecast Methods for Terrorist Events in Urban Environments. (*Presentation at the Second Symposium on Intelligence and Security Informatics*, (2004).
- [4] Brown, D.E., Hua Liu. Criminal Incident Prediction Using a Point-Pattern-Based Density Model. *International Journal of Forecasting*, v 19, 603-622. (2003).
- [5] Brown, D.E., Yifei Xue. Spatial Analysis with Preference Specification of Latent Decision Makers for Criminal Event Prediction. *Decision Support systems*, v 41, 560-573. (2006).
- [6] Dalton, J., Prats, F. Charlottesville Police Manpower Assessment Study, Spring 2005, Department of Systems and Information Engineering, University of Virginia.