Learning to Detect Patterns of Crime

Tong Wang[1], Cynthia Rudin[1], Daniel Wagner[2], and Rich Sevieri[2]

[1] Massachusetts Institute of Technology, Cambridge, MA 02139, USA
{tongwang,rudin}@mit.edu
[2] Cambridge Police Department, Cambridge, MA 02139, USA
{dwagner,rsevieri}@cambridgepolice.org

Abstract. Our goal is to automatically detect patterns of crime. Among a large set of crimes that happen every year in a major city, it is challenging, time-consuming, and labor-intensive for crime analysts to determine which ones may have been committed by the same individual(s). If automated, data-driven tools for crime pattern detection are made available to assist analysts, these tools could help police to better understand patterns of crime, leading to more precise attribution of past crimes, and the apprehension of suspects. To do this, we propose a pattern detection algorithm called *Series Finder*, that grows a pattern of discovered crimes from within a database, starting from a "seed" of a few crimes. Series Finder incorporates both the common characteristics of all patterns and the unique aspects of each specific pattern, and has had promising results on a decade's worth of crime pattern data collected by the Crime Analysis Unit of the Cambridge Police Department.

Keywords: Pattern detection, crime data mining, predictive policing.

1 Introduction

The goal of crime data mining is to understand patterns in criminal behavior in order to predict crime, anticipate criminal activity and prevent it (e.g., see [1]). There is a recent movement in law enforcement towards more empirical, data driven approaches to *predictive policing*, and the National Institute of Justice has recently launched an initiative in support of predictive policing [2]. However, even with new data-driven approaches to crime prediction, the fundamental job of crime analysts still remains difficult and often manual; *specific* patterns of crime are not necessarily easy to find by way of automated tools, whereas larger-scale density-based trends comprised mainly of background crime levels are much easier for data-driven approaches and software to estimate. The most frequent (and most successful) method to identify specific crime patterns involves the review of crime reports each day and the comparison of those reports to past crimes [3], even though this process can be extraordinarily time-consuming. In making these comparisons, an analyst looks for enough commonalities between a past crime and a present crime to suggest a pattern. Even though automated detection of specific crime patterns can be a much more difficult problem than

H. Blockeel et al. (Eds.): ECML PKDD 2013, Part III, LNAI 8190, pp. 515–530, 2013.

estimating background crime levels, tools for solving this problem could be extremely valuable in assisting crime analysts, and could directly lead to actionable preventative measures. Locating these patterns automatically is a challenge that machine learning tools and data mining analysis may be able to handle in a way that directly complements the work of human crime analysts.

In this work, we take a machine learning approach to the problem of detecting specific patterns of crime that are committed by the same offender or group. Our learning algorithm processes information similarly to how crime analysts process information instinctively: the algorithm searches through the database looking for similarities between crimes in a growing pattern and in the rest of the database, and tries to identify the modus operandi (M.O.) of the particular offender. The M.O. is the set of habits that the offender follows, and is a type of motif used to characterize the pattern. As more crimes are added to the set, the M.O. becomes more well-defined. Our approach to pattern discovery captures several important aspects of patterns:

- *Each M.O. is different.* Criminals are somewhat self-consistent in the way they commit crimes. However, different criminals can have very different M.O.'s. Consider the problem of predicting housebreaks (break-ins): Some offenders operate during weekdays while the residents are at work; some operate stealthily at night, while the residents are sleeping. Some offenders favor large apartment buildings, where they can break into multiple units in one day; others favor single-family houses, where they might be able to steal more valuable items. Different combinations of crime attributes can be more important than others for characterizing different M.O's.
- *General commonalities in M.O. do exist.* Each pattern is different but, for instance, similarity in time and space are often important to any pattern and should generally by weighted highly. Our method incorporates both general trends in M.O. and also pattern-specific trends.
- *Patterns can be dynamic.* Sometimes the M.O. shifts during a pattern. For instance, a novice burglar might initially use bodily force to open a door. As he gains experience, he might bring a tool with him to pry the door open. Occasionally, offenders switch entirely from one neighborhood to another. Methods that consider an M.O. as stationary would not naturally be able to capture these dynamics.

2 Background and Related Work

In this work, we define a "pattern" as a series of crimes committed by the same offender or group of offenders. This is different from a "hotspot" which is a spatially localized area where many crimes occur, whether or not they are committed by the same offender. It is also different than a "near-repeat" effect which is localized in time and space, and does not require the crimes to be committed by the same offender. To identify true patterns, one would need to consider information beyond simply time and space, but also other features of the crimes, such as the type of premise and means of entry.

An example of a pattern of crime would be a series of housebreaks over the course of a season committed by the same person, around different parts of East Cambridge, in houses whose doors are left unlocked, between noon and 2pm on weekdays. For this pattern, sometimes the houses are ransacked and sometimes not, and sometimes the residents are inside and sometimes not. This pattern does not constitute a "hotspot" as it's not localized in space. These crimes are not "near-repeats" as they are not localized in time and space (see [4]).

We know of very few previous works aimed directly at detecting specific patterns of crime. One of these works is that of Dahbur and Muscarello [5],[1] who use a cascaded network of Kohonen neural networks followed by heuristic processing of the network outputs. However, feature grouping in the first step makes an implicit assumption that attributes manually selected to group together have the same importance, which is not necessarily the case: each crime series has a signature set of factors that are important for that specific series, which is one of the main points we highlighted in the introduction. Their method has serious flaws, for instance that crimes occurring before midnight and after midnight cannot be grouped together by the neural network regardless of how many similarities exists between them, hence the need for heuristics. Series Finder has no such serious modeling defect. Nath [6] uses a semi-supervised clustering algorithm to detect crime patterns. He developed a weighting scheme for attributes, but the weights are provided by detectives instead of learned from data, similar to the baseline comparison methods we use. Brown and Hagen [7] and Lin and Brown [8] use similarity metrics like we do, but do not learn parameters from past data.

Many classic data mining techniques have been successful for crime analysis generally, such as association rule mining [7–10], classification [11], and clustering [6]. We refer to the general overview of Chen et al. [12], in which the authors present a general framework for crime data mining, where many of these standard tools are available as part of the COPLINK [13] software package. Much recent work has focused on locating and studying hotspots, which are localized high-crime-density areas (e.g., [14–16], and for a review, see [17]).

Algorithmic work on semi-supervised clustering methods (e.g., [18, 19]) is slightly related to our approach, in the sense that the set of patterns previously labeled by the police can be used as constraints for learned clusters; on the other hand, each of our clusters has different properties corresponding to different M.O.'s, and most of the crimes in our database are not part of a pattern and do not belong to a cluster. Standard clustering methods that assume all points in a cluster are close to the cluster center would also not be appropriate for modeling dynamic patterns of crime. Also not suitable are clustering methods that use the same distance metric for different clusters, as this would ignore the pattern's M.O. Clustering is usually unsupervised, whereas our method is supervised. Work on (unsupervised) set expansion in information retrieval (e.g., [20, 21]) is very relevant to ours. In set expansion, they (like us) start with

[1] Also see `http://en.wikipedia.org/wiki/Classification_System_for_Serial_Criminal_Patterns`

a small seed of instances, possess a sea of unlabeled entities (webpages), most of which are not relevant, and attempt to grow members of the same set as the seed. The algorithms for set expansion do not adapt to the set as it develops, which is important for crime pattern detection. The baseline algorithms we compare with are similar to methods like Bayesian Sets applied in the context of Growing a List [20, 21] in that they use a type of inner product as the distance metric.

3 Series Finder for Pattern Detection

Series Finder is a supervised learning method for detecting patterns of crime. It has two different types of coefficients: pattern-specific coefficients $\{\eta_{\hat{\mathcal{P}},j}\}_j$, and pattern-general coefficients $\{\lambda_j\}_j$. The attributes of each crime (indexed by j) capture elements of the M.O. such as the means of entry, time of day, etc. Patterns of crime are grown sequentially, starting from candidate crimes (the seed). As the pattern grows, the method adapts the pattern-specific coefficients in order to better capture the M.O. The algorithm stops when there are no more crimes within the database that are closely related to the pattern.

The crime-general coefficients are able to capture common characteristics of all patterns (bullet 2 in the introduction), and the pattern-specific coefficients adjust to each pattern's M.O. (bullet 1 in the introduction). Dynamically changing patterns (bullet 3 in the introduction) are captured by a similarity S, possessing a parameter d which controls the "degree of dynamics" of a pattern. We discuss the details within this section.

Let us define the following:

- $\mathscr{C}$ – A set of all crimes.
- $\mathscr{P}$ – A set of all patterns.
- $\mathcal{P}$ – A single pattern, which is a set of crimes. $\mathcal{P} \subseteq \mathscr{P}$.
- $\hat{\mathcal{P}}$ – A pattern grown from a seed of pattern $\mathcal{P}$. Ideally, if $\mathcal{P}$ is a true pattern and $\hat{\mathcal{P}}$ is a discovered pattern, then $\hat{\mathcal{P}}$ should equal $\mathcal{P}$ when it has been completed. Crimes in $\hat{P}$ are represented by $C_1, C_2, ...C_{|\hat{\mathcal{P}}|}$.
- $\mathcal{C}_{\hat{\mathcal{P}}}$ – The set of candidate crimes we will consider when starting from $\hat{\mathcal{P}}$ as the seed. These are potentially part of pattern $\mathcal{P}$. In practice, $\mathcal{C}_{\hat{\mathcal{P}}}$ is usually a set of crimes occurring within a year of the seed of $\mathcal{P}$. $\mathcal{C}_{\hat{\mathcal{P}}} \subseteq \mathscr{C}$.
- $s_j(C_i, C_k)$ – Similarity between crime i and k in attribute j. There are a total of J attributes. These similarities are calculated from raw data.
- $\gamma_{\hat{\mathcal{P}}}(C_i, C_k)$ – The overall similarity between crime i and k. It is a weighted sum of all J attributes, and is pattern-specific.

3.1 Crime-Crime Similarity

The pairwise similarity γ measures how similar crimes C_i and C_k are in a pattern set $\hat{\mathcal{P}}$. We model it in the following form:

$$\gamma_{\hat{\mathcal{P}}}(C_i, C_k) = \frac{1}{\Gamma_{\hat{\mathcal{P}}}} \sum_{j=1}^{J} \lambda_j \eta_{\hat{\mathcal{P}},j} s_j(C_i, C_k),$$

where two types of coefficients are introduced:

1. λ_j – pattern-general weights. These weights consider the general importance of each attribute. They are trained on past patterns of crime that were previously labeled by crime analysts as discussed in Section 3.4.
2. $\eta_{\hat{\mathcal{P}},j}$ – pattern-specific weights. These weights capture characteristics of a specific pattern. All crimes in pattern $\hat{\mathcal{P}}$ are used to decide $\eta_{\hat{\mathcal{P}},j}$, and further, the defining characteristics of $\hat{\mathcal{P}}$ are assigned higher values. Specifically:

$$\eta_{\hat{\mathcal{P}},j} = \sum_{i=1}^{|\hat{\mathcal{P}}|} \sum_{k=1}^{|\hat{\mathcal{P}}|} s_j(C_i, C_k)$$

$\Gamma_{\hat{\mathcal{P}}}$ is the normalizing factor $\Gamma_{\hat{\mathcal{P}}} = \sum_{j=1}^{J} \lambda_j \eta_{\hat{\mathcal{P}},j}$. Two crimes have a high $\gamma_{\hat{\mathcal{P}}}$ if they are similar along attributes that are important specifically to that crime pattern, and generally to all patterns.

3.2 Pattern-Crime Similarity

Pattern-crime similarity S measures whether crime $\tilde{C}$ is similar enough to set $\hat{\mathcal{P}}$ that it should be potentially included in $\hat{\mathcal{P}}$. The pattern-crime similarity incorporates the dynamics in M.O. discussed in the introduction. The dynamic element is controlled by a parameter d, called the *degree of dynamics*. The pattern-crime similarity is defined as follows for pattern $\hat{\mathcal{P}}$ and crime $\tilde{C}$:

$$S(\hat{\mathcal{P}}, \tilde{C}) = \left(\frac{1}{|\hat{\mathcal{P}}|} \sum_{i=1}^{|\hat{\mathcal{P}}|} \gamma_{\hat{\mathcal{P}}}(\tilde{C}, C_i)^d \right)^{(1/d)}$$

where $d \geqslant 1$. This is a soft-max, that is, an ℓ_d norm over $i \in \hat{\mathcal{P}}$. Use of the soft-max allows the pattern $\hat{\mathcal{P}}$ to evolve: crime i needs only be very similar to a few crimes in $\hat{\mathcal{P}}$ to be considered for inclusion in $\hat{\mathcal{P}}$ when the degree of dynamics d is large. On the contrary, if d is 1, this forces patterns to be very stable and stationary, as new crimes would need to be similar to most or all of the crimes already in $\hat{\mathcal{P}}$ to be included. That is, if $d = 1$, the dynamics of the pattern are ignored. For our purpose, d is chosen appropriately to balance between including the dynamics (d large), and stability and compactness of the pattern (d small).

3.3 Sequential Pattern Building

Starting with the seed, crimes are added iteratively from $\mathcal{C}_{\hat{\mathcal{P}}}$ to $\hat{\mathcal{P}}$. At each iteration, the candidate crime with the highest pattern-crime similarity to $\hat{\mathcal{P}}$ is tentatively added to $\hat{\mathcal{P}}$. Then $\hat{\mathcal{P}}$'s cohesion is evaluated, which measures the cohesiveness of $\hat{\mathcal{P}}$ as a pattern of crime: $\text{Cohesion}(\hat{\mathcal{P}}) = \frac{1}{|\hat{\mathcal{P}}|} \sum_{i \in \hat{\mathcal{P}}} S(\hat{\mathcal{P}} \backslash C_i, C_i)$. While the cohesion is large enough, we will proceed to grow $\hat{\mathcal{P}}$. If $\hat{\mathcal{P}}$'s cohesion is below a threshold, $\hat{\mathcal{P}}$ stops growing. Here is the formal algorithm:

1: **Initialization**: $\hat{\mathcal{P}} \leftarrow$ {Seed crimes}
2: **repeat**
3: $C_{\text{tentative}} = \arg\max_{C \in (\mathcal{C}_{\hat{\mathcal{P}}} \setminus \hat{\mathcal{P}})} S(\hat{\mathcal{P}}, C)$
4: $\hat{\mathcal{P}} \leftarrow \hat{\mathcal{P}} \cup \{C_{\text{tentative}}\}$
5: Update: $\eta_{\hat{\mathcal{P}},j}$ for $j \in \{1, 2, \ldots J\}$, and Cohesion($\hat{\mathcal{P}}$)
6: **until** Cohesion($\hat{\mathcal{P}}$) < cutoff
7: $\hat{\mathcal{P}}^{\text{final}} := \hat{\mathcal{P}} \backslash C_{\text{tentative}}$
8: **return** $\hat{\mathcal{P}}^{\text{final}}$

3.4 Learning the Pattern-General Weights $\boldsymbol{\lambda}$

The pattern-general weights are trained on past pattern data, by optimizing a performance measure that is close to the performance measures we will use to evaluate the quality of the results. Note that an alternative approach would be to simply ask crime analysts what the optimal weighting should be, which was the approach taken by Nath [6]. (This simpler method will also be used in Section 5.2 as a baseline for comparison.) We care fundamentally about optimizing the following measures of quality for our returned results:

– The fraction of the true pattern $\mathcal{P}$ returned by the algorithm:

$$\text{Recall}(\mathcal{P}, \hat{\mathcal{P}}) = \frac{\sum_{C \in \mathcal{P}} \mathbb{1}(C \in \hat{\mathcal{P}})}{|\mathcal{P}|}.$$

– The fraction of the discovered crimes that are within pattern $\mathcal{P}$:

$$\text{Precision}(\mathcal{P}, \hat{\mathcal{P}}) = \frac{\sum_{C \in \hat{\mathcal{P}}} \mathbb{1}(C \in \mathcal{P})}{|\hat{\mathcal{P}}|}.$$

The training set consists of true patterns $\mathcal{P}_1, \mathcal{P}_2, \ldots \mathcal{P}_\ell, \ldots \mathcal{P}_{|\mathscr{P}|}$. For each pattern $\mathcal{P}_\ell$ and its corresponding $\hat{\mathcal{P}}_\ell$, we define a gain function $g(\hat{\mathcal{P}}_\ell, \mathcal{P}_\ell, \boldsymbol{\lambda})$ containing both precision and recall. The dependence on $\boldsymbol{\lambda} = \{\lambda_j\}_{j=1}^J$ is implicit, as it was used to construct $\hat{\mathcal{P}}_\ell$.

$$g(\hat{\mathcal{P}}_\ell, \mathcal{P}_\ell, \boldsymbol{\lambda}) = \text{Recall}(\mathcal{P}_\ell, \hat{\mathcal{P}}_\ell) + \beta \cdot \text{Precision}(\mathcal{P}_\ell, \hat{\mathcal{P}}_\ell)$$

where β is the trade-off coefficient between the two quality measures. We wish to choose $\boldsymbol{\lambda}$ to maximize the gain over all patterns in the training set.

$$\begin{aligned} \underset{\boldsymbol{\lambda}}{\text{maximize}} \quad & G(\boldsymbol{\lambda}) = \sum_\ell g(\hat{\mathcal{P}}_\ell, \mathcal{P}_\ell, \boldsymbol{\lambda}) \\ \text{subject to} \quad & \lambda_j \geqslant 0,\ j = 1, \ldots, J, \\ & \sum_{j=1}^{J} \lambda_j = 1. \end{aligned}$$

The optimization problem is non-convex and non-linear. However we hypothesize that it is reasonably smooth: small changes in $\boldsymbol{\lambda}$ translate to small changes

in G. We use coordinate ascent to approximately optimize the objective, starting from different random initial conditions to avoid returning local minima. The procedure works as follows:

Initialize $\boldsymbol{\lambda}$ randomly, Converged=0
while Converged=0 **do**
  for $j = 1 \rightarrow J$ **do**
    $\lambda_j^{\text{new}} \leftarrow \text{argmax}_{\lambda_j}\, G(\boldsymbol{\lambda})$ (using a linesearch for the optima)
  end for
  if $\boldsymbol{\lambda}^{\text{new}} = \boldsymbol{\lambda}$ **then**
    Converged= 1
  else
    $\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda}^{\text{new}}$
  end if
end while
return $\boldsymbol{\lambda}$

We now discuss the definition of each of the J similarity measures.

4 Attribute Similarity Measures

Each pairwise attribute similarity $s_j : \mathcal{C} \times \mathcal{C} \rightarrow [0, 1]$ compares two crimes along attribute j. Attributes are either categorical or numerical, and by the nature of our data, we are required to design similarity measures of both kinds.

4.1 Similarity for Categorical Attributes

In the Cambridge Police database for housebreaks, categorical attributes include "type of premise" (apartment, single-family house, etc.), "ransacked" (indicating whether the house was ransacked) and several others. We wanted a measure of agreement between crimes for each categorical attribute that includes (i) whether the two crimes agree on the attribute (ii) how common that attribute is. If the crimes do not agree, the similarity is zero. If the crimes do agree, and agreement on that attribute is unusual, the similarity should be given a higher weight. For example, in residential burglaries, it is unusual for the resident to be at home during the burglary. Two crimes committed while the resident was in the home are more similar to each other than two crimes where the resident was not at home. To do this, we weight the similarity by the probability of the match occurring, as follows, denoting C_{ij} as the jth attribute for crime C_i:

$$s_j(C_i, C_k) = \begin{cases} 1 - \sum_{q \in Q} p_j^2(x) & \text{if} \quad C_{ij} = C_{kj} = x \\ 0 & \text{if} \quad C_{ij} \neq C_{kj} \end{cases}$$

where $p_j^2(x) = \frac{n_x(n_x-1)}{N(N-1)}$, with n_x the number of times x is observed in the collection of N crimes. This is a simplified version of Goodall's measure [22].

4.2 Similarity for Numerical Attributes

Two formats of data exist for numerical attributes, either exact values, such as time 3:26pm, or a time window, e.g., 9:45am - 4:30pm. Unlike other types of crime such as assault and street robbery, housebreaks usually happen when the resident is not present, and thus time windows are typical. In this case, we need a similarity measure that can handle both exact time information and range-of-time information. A simple way of dealing with a time window is to take the midpoint of it (e.g., [15]), which simplifies the problem but may introduce bias.

Time-of-day profiles. We divide data into two groups: exact data $(t_1, t_2, \ldots, t_{m_e})$, and time window data $(\tilde{t}_1, \tilde{t}_2, \ldots, \tilde{t}_{m_r})$ where each data point is a range, $\tilde{t}_i = [t_{i,1}, t_{i,2}]$, $i = 1, 2, \ldots m_r$. We first create a profile based only on crimes with exact time data using kernel density estimation: $\hat{p}_{\text{exact}}(t) \propto \frac{1}{m_e} \sum_{i=1}^{m_e} K(t - t_i)$ where the kernel $K(\cdot)$ is a symmetric function, in our case a gaussian with a chosen bandwidth (we chose one hour). Then we use this to obtain an approximate distribution incorporating the time window measurements, as follows:

$$\begin{aligned} p(t|\tilde{t}_1, \ldots, \tilde{t}_{m_r}) &\propto p(t) \cdot p(\tilde{t}_1, \ldots, \tilde{t}_{m_r}|t) \\ &\approx \hat{p}_{\text{exact}}(t) \cdot \hat{p}(\text{range includes } t|t). \end{aligned}$$

The function $\hat{p}(\text{range includes } t|t)$ is a smoothed version of the empirical probability that the window includes t:

$$\hat{p}(\text{range includes } t|t) \propto \frac{1}{m_r} \sum_{i=1}^{m_r} \tilde{K}(t, \tilde{t}_i)$$

where $\tilde{t}_i = [t_{i,1}, t_{i,2}]$ and $\tilde{K}(t, \tilde{t}_i) := \int_\tau \mathbf{1}_{\tau \in [t_{i,1}, t_{i,2}]} K(t - \tau) \mathrm{d}\tau$. K is again a gaussian with a selected bandwidth. Thus, we define:

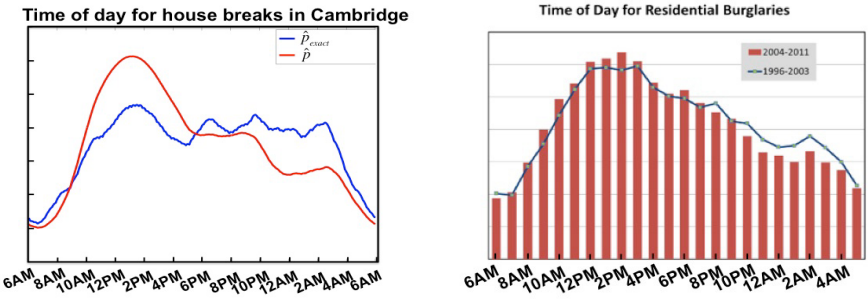
$$\hat{p}_{\text{range}}(t) \propto \hat{p}_{\text{exact}}(t) \cdot \frac{1}{m_r} \sum_{i=1}^{m_r} \tilde{K}(t, \tilde{t}_i).$$

We combine the exact and range estimates in a weighted linear combination, weighted according to the amount of data we have from each category:

$$\hat{p}(t) \propto \frac{m_e}{m_e + m_r} \hat{p}_{\text{exact}}(t) + \frac{m_r}{m_e + m_r} \hat{p}_{\text{range}}(t).$$

We used the approach above to construct a time-of-day profile for residential burglaries in Cambridge, where $\hat{p}(t)$ and $\hat{p}_{\text{exact}}(t)$ are plotted in Figure 1(a). To independently verify the result, we compared it with residential burglaries in Portland between 1996 and 2011 (reproduced from [23]) shown in Figure 1(b).[2] The temporal pattern is similar, with a peak at around 1-2pm, a drop centered around 6-7am, and a smaller drop at around midnight, though the profile differs slightly in the evening between 6pm-2am.

[2] To design this plot for Portland, range-of-time information was incorporated by distributing the weight of each crime uniformly over its time window.



(a) Cambridge, from year 1997 to 2012 (b) Portland, from year 1996 to 2011

Fig. 1. Time of day profiling for house breaks in two different cities

A unified similarity measure for numeric attributes. We propose a similarity measure that is consistent for exact and range numerical data. The similarity decays exponentially with the distance between two data values, for either exact or range data. We use the expected average distance over the two ranges as the distance measure. For example, let crime i happen within $t_i := [t_{i,1}, t_{i,2}]$ and crime k happen within $t_k := [t_{k,1}, t_{k,2}]$. Then

$$\tilde{d}(t_i, t_k) = \int_{t_{i,1}}^{t_{i,2}} \int_{t_{k,1}}^{t_{k,2}} \hat{p}(\tau_i|t_i)\hat{p}(\tau_k|t_k)d(\tau_i, \tau_k)\mathrm{d}\tau_i\mathrm{d}\tau_k$$

where $\hat{p}$ was estimated in the previous subsection for times of the day, and $d(\tau_i, \tau_k)$ is the difference in time between τ_i and τ_k. The conditional probability is obtained by renormalizing $\hat{p}(\tau_i|t_i)$ to the interval (or exact value) t_i. The distance measure for exact numerical data can be viewed as a special case of this expected average distance where the conditional probability $\hat{p}(\tau_i|t_i)$ is 1.

The general similarity measure is thus:

$$s_j(C_i, C_k) := \exp\left(-\tilde{d}(z_i, z_k)/\Upsilon_j\right)$$

where Υ_j is a scaling factor (e.g, we chose $\Upsilon_j = 120$ minutes in the experiment), and z_i, z_k are values of attribute j for crimes i and k, which could be either exact values or ranges of values. We applied this form of similarity measure for all numerical (non-categorical) crime attributes.

5 Experiments

We used data from 4855 housebreaks in Cambridge between 1997 and 2006 recorded by the Crime Analysis Unit of the Cambridge Police Department. Crime attributes include geographic location, date, day of the week, time frame, location of entry, means of entry, an indicator for "ransacked," type of premise, an

indicator for whether residents were present, and suspect and victim information. We also have 51 patterns collected over the same period of time that were curated and hand-labeled by crime analysts.

5.1 Evaluation Metrics

The evaluation metrics used for the experimental results are *average precision* and *reciprocal rank*. Denoting $\hat{\mathcal{P}}^i$ as the first i crimes in the discovered pattern, and $\Delta\text{Recall}(\mathcal{P}, \hat{\mathcal{P}}^i)$ as the change in recall from $i-1$ to i:

$$\text{AveP}(\mathcal{P}, \hat{\mathcal{P}}) := \sum_{i=1}^{|\hat{\mathcal{P}}|} \text{Precision}(\mathcal{P}, \hat{\mathcal{P}}^i) \Delta\text{Recall}(\mathcal{P}, \hat{\mathcal{P}}^i).$$

To calculate reciprocal rank, again we index the crimes in $\hat{\mathcal{P}}$ by the order in which they were discovered, and compute

$$\text{RR}(\mathcal{P}, \hat{\mathcal{P}}) := \frac{1}{\left(\sum_{r=1}^{|\hat{\mathcal{P}}|} \frac{1}{r}\right)} \sum_{C_i \in \mathcal{P}} \frac{1}{\text{Rank}(C_i, \hat{\mathcal{P}})},$$

where $\text{Rank}(C_i, \hat{\mathcal{P}})$ is the order in which C_i was added to $\hat{\mathcal{P}}$. If C_i was never added to $\hat{\mathcal{P}}$, then $\text{Rank}(C_i, \hat{\mathcal{P}})$ is infinity and the term in the sum is zero.

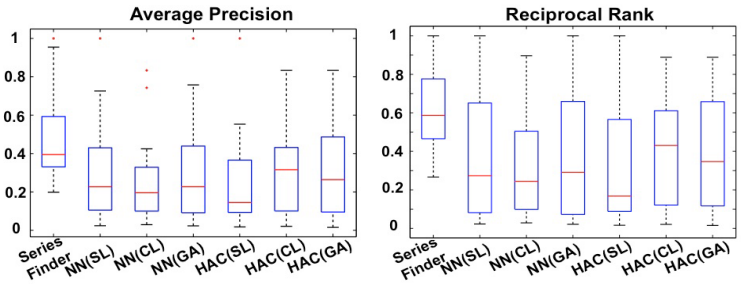
5.2 Competing Models and Baselines

We compare with hierarchical agglomerative clustering and an iterative nearest neighbor approach as competing baseline methods. For each method, we use several different schemes to iteratively add discovered crimes, starting from the same seed given to Series Finder. The pairwise similarity γ is a weighted sum of the attribute similarities:

$$\gamma(C_i, C_k) = \sum_{j=1}^{J} \hat{\lambda}_j s_j(C_i, C_k).$$

where the similarity metrics $s_j(C_i, C_k)$ are the same as Series Finder used. The weights $\hat{\boldsymbol{\lambda}}$ were provided by the Crime Analysis Unit of the Cambridge Police Department based on their experience. This will allow us to see the specific advantage of Series Finder, where the weights were learned from past data.

Hierarchical agglomerative clustering (HAC) begins with each crime as a singleton cluster. At each step, the most similar (according to the similarity criterion) two clusters are merged into a single cluster, producing one less cluster at the next level. *Iterative nearest neighbor classification* (NN) begins with the seed set. At each step, the nearest neighbor (according to the similarity criterion) of the set is added to the pattern, until the nearest neighbor is no longer sufficiently similar. HAC and NN were used with three different criteria for cluster-cluster or cluster-crime similarity: *Single Linkage* (SL), which considers the most similar



(a) (b)

Fig. 2. Boxplot of evaluation metrics for out-of-sample patterns

pair of crimes; *Complete Linkage* (CL), which considers the most dissimilar pair of crimes, and *Group Average* (GA), which uses the averaged pairwise similarity [24]. The incremental nearest neighbor algorithm using the S_{GA} measure, with the weights provided by the crime analysts, becomes similar in spirit to the Bayesian Sets algorithm [20] and how it is used in information retrieval applications [21].

$$S_{SL}(R,T) := \max_{C_i \in R, C_k \in T} \gamma(C_i, C_k)$$
$$S_{CL}(R,T) := \min_{C_i \in R, C_k \in T} \gamma(C_i, C_k)$$
$$S_{GA}(R,T) := \frac{1}{|R||T|} \sum_{C_i \in R} \sum_{C_k \in T} \gamma(C_i, C_k).$$

5.3 Testing

We trained our models on two-thirds of the patterns from the Cambridge Police Department and tested the results on the remaining third. For all methods, pattern $\hat{\mathcal{P}}_\ell$ was grown until all crimes in $\mathcal{P}_\ell$ were discovered. Boxplots of the distribution of average precision and reciprocal ranks over the test patterns for Series Finder and six baselines are shown in Figure 2(a) and Figure 2(b). We remark that Series Finder has several advantages over the competing models: (i) Hierarchical agglomerative clustering does not use the similarity between seed crimes. Each seed grows a pattern independently, with possibly no interaction between seeds. (ii) The competing models do not have pattern-specific weights. One set of weights, which is pattern-general, is used for all patterns. (iii) The weights used by the competing models are provided by detectives based on their experience, while the weights of Series Finder are learned from data.

Since Series Finder's performance depends on pattern-specific weights that are calculated from seed crimes, we would like to understand how much each additional crime within the seed generally contributes to performance. The average precision and reciprocal rank for the 16 testing patterns grown from 2, 3

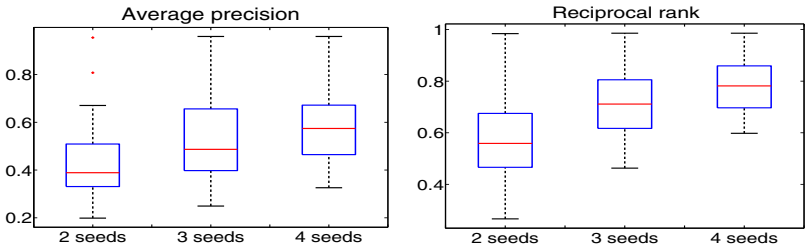


Fig. 3. Performance of Series Finder with 2, 3 and 4 seeds

and 4 seeds are plotted in Figure 3(a) and Figure 3(b). For both performance measures, the quality of the predictions increases consistently with the number of seed crimes. The additional crimes in the seed help to clarify the M.O.

5.4 Model Convergence and Sensitivity Analysis

In Section 3, when discussing the optimization procedure for learning the weights, we hypothesized that small changes in $\boldsymbol{\lambda}$ generally translate to small changes in the objective $G(\boldsymbol{\lambda})$. Our observations about convergence have been consistent with this hypothesis, in that the objective seems to change smoothly over the course of the optimization procedure. Figure 4(a) shows the optimal objective value at each iteration of training the algorithm on patterns collected by the Cambridge Police Department. In this run, convergence was achieved after 14 coordinate ascent iterations. This was the fastest converging run over the randomly chosen initial conditions used for the optimization procedure.

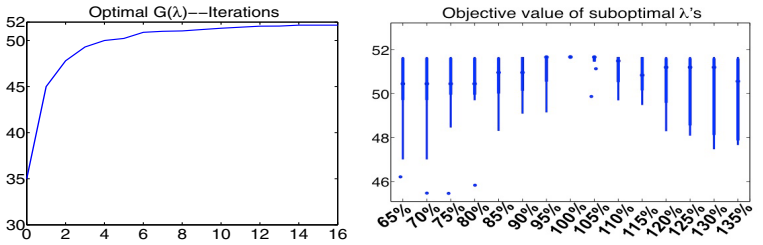


Fig. 4. Performance analysis

We also performed a sensitivity analysis for the optimum. We varied each of the J coefficients λ_j from 65% to 135%, of its value at the optimum. As each coefficient was varied, the others were kept fixed. We recorded the value of $G(\boldsymbol{\lambda})$

at several points along this spectrum of percentages between 65% and 135%, for each of the λ_j's. This allows us to understand the sensitivity of $G(\boldsymbol{\lambda})$ to movement along any one of the axes of the J-dimensional space. We created box plots of $G(\boldsymbol{\lambda})$ at every 5th percentage between between 65% and 135%, shown in Figure 4(b). The number of elements in each box plot is the number of dimensions J. These plots provide additional evidence that the objective $G(\boldsymbol{\lambda})$ is somewhat smooth in $\boldsymbol{\lambda}$; for instance the objective value varies by a maximum of approximately 5-6% when one of the λ_j's changes by 10-15%.

6 Expert Validation and Case Study

We wanted to see whether our data mining efforts could help crime analysts identify crimes within a pattern that they did not yet know about, or exclude crimes that were misidentified as part of a pattern. To do this, Series Finder was trained on all existing crime patterns from the database to get the pattern-general weights $\boldsymbol{\lambda}$. Next, using two crimes in each pattern as a seed, Series Finder iteratively added candidate crimes to the pattern until the pattern cohesion dropped below 0.8 of the seed cohesion. Crime analysts then provided feedback on Series Finder's results for nine patterns.

There are now three versions of each pattern: $\mathcal{P}$ which is the original pattern in the database, $\hat{\mathcal{P}}$ which was discovered using Series Finder from two crimes in the pattern, and $\mathcal{P}_{\text{verified}}$ which came from crime experts after they viewed the union of $\hat{\mathcal{P}}$ and $\mathcal{P}$. Based on these, we counted different types of successes and failures for the 9 patterns, shown in Table 1. The mathematical definition of them is represented by the first 4 columns. For example, *correct finds* refer to crimes that are not in $\mathcal{P}$, but that are in $\hat{\mathcal{P}}$, and were verified by experts as belonging to the pattern, in $\mathcal{P}_{\text{verified}}$.

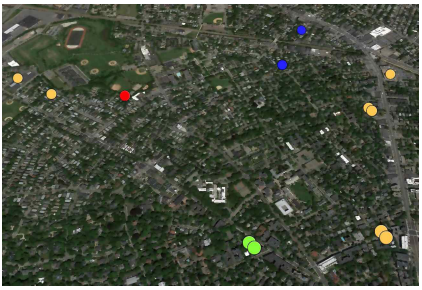
Table 1. Expert validation study results

Type of crimes	$\mathcal{P}$	$\hat{\mathcal{P}}$	$\mathcal{P}_{\text{verified}}$	$\mathcal{P}_1$	$\mathcal{P}_2$	$\mathcal{P}_3$	$\mathcal{P}_4$	$\mathcal{P}_5$	$\mathcal{P}_6$	$\mathcal{P}_7$	$\mathcal{P}_8$	$\mathcal{P}_9$
Correct hits	$\subseteq$	$\subseteq$	$\subseteq$	6	5	6	3	8	5	7	2	10
Correct finds	$\notin$	$\subseteq$	$\subseteq$	2	1	0	1	0	1	2	2	0
Correct exclusions	$\subseteq$	$\notin$	$\notin$	0	0	4	1	0	2	1	0	0
Incorrect exclusions	$\subseteq$	$\notin$	$\subseteq$	0	0	1	0	1	0	0	0	1
False hits	$\notin$	$\subseteq$	$\notin$	2	0	0	0	2	2	0	0	0

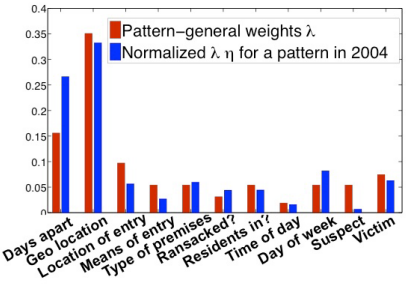
Correct hits, *correct finds* and *correct exclusions* count successes for Series Finder. Specifically, correct finds and correct exclusions capture Series Finder's improvements over the original database. Series Finder was able to discover 9 crimes that analysts had not previously matched to a pattern (the sum of the correct finds) and exclude 8 crimes that analysts agreed should be excluded (the sum of correct exclusions). *Incorrect exclusions* and *false hits* are not successes.

Table 2. Example: A 2004 Series

NO	Cri type	Date	Loc of entry	Mns of entry	Premises	Rans	Resid	Time of day	Day	Suspect	Victim
1	Seed	1/7/04	Front door	Pried	Aptment	No	Not in	8:45	Wed	null	White F
2	Corr hit	1/18/04	Rear door	Pried	Aptment	Yes	Not in	12:00	Sun	White M	White F
3	Corr hit	1/26/04	Grd window	Removed	Res Unk	No	Not in	7:30-12:15	Mon	null	Hisp F
4	Seed	1/27/04	Rear door	Popped Lock	Aptment	No	Not in	8:30-18:00	Tues	null	null
5	Corr exclu	1/31/04	Grd window	Pried	Res Unk	No	Not in	13:21	Sat	Black M	null
6	Corr hit	2/11/04	Front door	Pried	Aptment	No	Not in	8:30-12:30	Wed	null	Asian M
7	Corr hit	2/11/04	Front door	Pried	Aptment	No	Not in	8:00-14:10	Wed	null	null
8	Corr hit	2/17/04	Grd window	Unknown	Aptment	No	Not in	0:35	Tues	null	null
9	Corr find	2/19/04	Door: unkn	Pried	Aptment	No	Not in	10:00-16:10	Thur	null	White M
10	Corr find	2/19/04	Door: unkn	Pried	Aptment	No	Not in	7:30-16:10	Thur	null	White M
11	Corr hit	2/20/04	Front door	Broke	Aptment	No	Not in	8:00-17:55	Fri	null	null
12	Corr hit	2/25/04	Front door	Pried	Aptment	Yes	Not in	14:00	Wed	null	null



(a) Locations of crimes



(b) λ and $\frac{1}{\Gamma_{\hat{\mathcal{P}}}}\lambda \cdot \eta$ for a pattern in 2004

Fig. 5. An example pattern in 2004

On the other hand, false hits that are similar to the crimes within the pattern may still be useful for crime analysts to consider when determining the M.O.

We now discuss a pattern in detail to demonstrate the type of result that Series Finder is producing. The example provided is Pattern 7 in Table 1, which is a series from 2004 in Mid-Cambridge covering a time range of two months. Crimes were usually committed on weekdays during working hours. The premises are all apartments (except two unknowns). Figure 5(a) shows geographically where these crimes were located. In Figure 5(a), four categories of crime within the 2004 pattern are marked with different colored dots: seed crimes are represented with blue dots, correct hits are represented with orange dots, the correct exclusion is represented with a red dot and the two correct finds are represented with green dots. Table 2 provides some details about the crimes within the series.

We visualize the M.O. of the pattern by displaying the weights in Figure 5(b). The red bars represent the pattern-general weights $\boldsymbol{\lambda}$ and the blue bars represent the total normalized weights obtained from the product of pattern-general weights and pattern-specific weights for this 2004 pattern. Notable observations about this pattern are that: the time between crimes is a (relatively) more important characteristic for this pattern than for general patterns, as the crimes in the pattern happen almost every week; the means and location of entry are relatively less important as they are not consistent; and the suspect information is also relatively less important. The suspect information is only present in one

of the crimes found by Series Finder (a white male). Geographic closeness is less important for this series, as the crimes in the series are spread over a relatively large geographic distance.

Series Finder made a contribution to this pattern, in the sense that it detected two crimes that analysts had not previously considered as belonging to this pattern. It also correctly excluded one crime from the series. In this case, the correct exclusion is valuable since it had suspect information, which in this case could be very misleading. This exclusion of this crime indicates that the offender is a white male, rather than a black male.

7 Conclusion

Series Finder is designed to detect patterns of crime committed by the same individual(s). In Cambridge, it has been able to correctly match several crimes to patterns that were originally missed by analysts. The designer of the near-repeat calculator, Ratcliffe, has stated that the near-repeat calculator is not a "silver bullet" [25]. Series Finder also is not a magic bullet. On the other hand, Series Finder can be a useful tool: by using very detailed information about the crimes, and by tailoring the weights of the attributes to the specific M.O. of the pattern, we are able to correctly pinpoint patterns more accurately than similar methods. As we have shown through examples, the extensive data processing and learning that goes into characterizing the M.O. of each pattern leads to richer insights that were not available previously. Some analysts spend hours each day searching for crime series manually. By replicating the cumbersome process that analysts currently use to find patterns, Series Finder could have enormous implications for time management, and may allow analysts to find patterns that they would not otherwise be able to find.

Acknowledgements. Funding for this work was provided by C. Rudin's grants from MIT Lincoln Laboratory and NSF-CAREER IIS-1053407. We wish to thank Christopher Bruce, Julie Schnobrich-Davis and Richard Berk for helpful discussions.

References

1. Berk, R., Sherman, L., Barnes, G., Kurtz, E., Ahlman, L.: Forecasting murder within a population of probationers and parolees: a high stakes application of statistical learning. Journal of the Royal Statistical Society: Series A (Statistics in Society) 172(1), 191–211 (2009)
2. Pearsall, B.: Predictive policing: The future of law enforcement? National Institute of Justice Journal 266, 16–19 (2010)
3. Gwinn, S.L., Bruce, C., Cooper, J.P., Hick, S.: Exploring crime analysis. Readings on essential skills, 2nd edn. BookSurge, LLC (2008)
4. Ratcliffe, J.H., Rengert, G.F.: Near-repeat patterns in Philadelphia shootings. Security Journal 21(1), 58–76 (2008)

5. Dahbur, K., Muscarello, T.: Classification system for serial criminal patterns. Artificial Intelligence and Law 11(4), 251–269 (2003)
6. Nath, S.V.: Crime pattern detection using data mining. In: Proceedings of Web Intelligence and Intelligent Agent Technology Workshops, pp. 41–44 (2006)
7. Brown, D.E., Hagen, S.: Data association methods with applications to law enforcement. Decision Support Systems 34(4), 369–378 (2003)
8. Lin, S., Brown, D.E.: An outlier-based data association method for linking criminal incidents. In: Proceedings of the Third SIAM International Conference on Data Mining. (2003)
9. Ng, V., Chan, S., Lau, D., Ying, C.M.: Incremental mining for temporal association rules for crime pattern discoveries. In: Proceedings of the 18th Australasian Database Conference, vol. 63, pp. 123–132 (2007)
10. Buczak, A.L., Gifford, C.M.: Fuzzy association rule mining for community crime pattern discovery. In: ACM SIGKDD Workshop on Intelligence and Security Informatics (2010)
11. Wang, G., Chen, H., Atabakhsh, H.: Automatically detecting deceptive criminal identities. Communications of the ACM 47(3), 70–76 (2004)
12. Chen, H., Chung, W., Xu, J., Wang, G., Qin, Y., Chau, M.: Crime data mining: a general framework and some examples. Computer 37(4), 50–56 (2004)
13. Hauck, R.V., Atabakhsb, H., Ongvasith, P., Gupta, H., Chen, H.: Using COPLINK to analyze criminal-justice data. Computer 35(3), 30–37 (2002)
14. Short, M.B., D'Orsogna, M.R., Pasour, V.B., Tita, G.E., Brantingham, P.J., Bertozzi, A.L., Chayes, L.B.: A statistical model of criminal behavior. Mathematical Models and Methods in Applied Sciences 18, 1249–1267 (2008)
15. Mohler, G.O., Short, M.B., Brantingham, P.J., Schoenberg, F.P., Tita, G.E.: Self-exciting point process modeling of crime. Journal of the American Statistical Association 106(493) (2011)
16. Short, M.B., D'Orsogna, M., Brantingham, P., Tita, G.: Measuring and modeling repeat and near-repeat burglary effects. Journal of Quantitative Criminology 25(3), 325–339 (2009)
17. Eck, J., Chainey, S., Cameron, J., Wilson, R.: Mapping crime: Understanding hotspots. Technical report, National Institute of Justice, NIJ Special Report (August 2005)
18. Basu, S., Banerjee, A., Mooney, R.: Semi-supervised clustering by seeding. In: International Conference on Machine Learning, pp. 19–26 (2002)
19. Wagstaff, K., Cardie, C., Rogers, S., Schrödl, S.: Constrained k-means clustering with background knowledge. In: Int'l Conf. on Machine Learning, pp. 577–584 (2001)
20. Ghahramani, Z., Heller, K.: Bayesian sets. In: Proceedings of Neural Information Processing Systems (2005)
21. Letham, B., Rudin, C., Heller, K.: Growing a list. Data Mining and Knowledge Discovery (to appear, 2013)
22. Boriah, S., Chandola, V., Kumar, V.: Similarity measures for categorical data: A comparative evaluation. In: Proceedings of the Eighth SIAM International Conference on Data Mining, pp. 243–254 (2008)
23. Criminal Justice Policy Research Institute: Residential burglary in Portland, Oregon. Hatfield School of Government, Criminal Justice Policy Research Institute, `http://www.pdx.edu/cjpri/time-of-dayday-of-week-0`
24. Hastie, T., Tibshirani, R., Friedman, J., Franklin, J.: The elements of statistical learning: data mining, inference and prediction. Springer (2005)
25. National Law Enforcement and Corrections Technology Center: 'Calculate' repeat crime. TechBeat (Fall 2008)