MIT Sloan School of Management

MIT Sloan School Working Paper 5073-14

Finding Patterns with a Rotten Core: Data Mining for Crime Series with Core Sets

Tong Wang, Cynthia Rudin,
Daniel Wagner, Rich Sevieri

Finding Patterns with a Rotten Core: Data Mining for Crime Series with Core Sets

Tong Wang and **Cynthia Rudin**
Massachusetts Institute of Technology
Cambridge, MA 02139, USA

Daniel Wagner and **Rich Sevieri**
Cambridge Police Department
Cambridge, MA 02139, USA

Abstract

One of the most challenging problems facing crime analysts is that of identifying "crime series" which are sets of crimes committed by the same individual or group. Detecting series' of crime can be an important step in predictive policing, as knowledge of an ongoing pattern can be of paramount importance towards stopping it. Currently, crime analysts detect patterns manually; our goal is to assist them by providing automated tools for discovering crime series from within a database of crimes. Our approach relies on a key hypothesis that each crime series possesses a *core* of crimes that are similar to each other, which can be used to characterize the modus operandi (M.O.) of the criminal. We find core sets of crime using an integer linear programming approach, and then construct the rest of the crime series by merging core sets to form the full crime series. To judge whether a crime series is indeed a core, we consider both *pattern-general similarity*, which can be learned from past crime series, and *pattern-specific similarity*, which is specific to the M.O. of the series and cannot be learned. We learn a similarity graph on the set of crimes to form the pattern-general similarity. Our method can be used for general pattern detection beyond crime series detection, as core sets exists for patterns in many domains.

Introduction

The foundation of *predictive policing* is that if we are able to predict crime, we can takes steps to prevent it. Empirical approaches to predictive policing have recently been adopted by many law enforcement agencies, and the National Institute of Justice has recently launched an initiative in support of predictive policing [Pearsall2010]. One of the most important problems in predictive policing is that of "crime series detection," or the detection of a set of crimes committed by the same individual or group. If a crime analyst can identify an ongoing crime series, it is possible that actions can be taken to stop this pattern, and possibly, to apprehend the suspect(s). Criminals follow a modus operandi (M.O.) that characterizes their crime series; for instance, some criminals operate exclusively during the day, others work at night, some criminals target apartments for housebreaks, while others target single family houses. It is a crime analysts' job to find these crime series, and they currently do this manually by identifying the M.O. characterizing the series [Gwinn et al.2008]. If we can use automated tools to detect crime series and identify the M.O., it could potentially assist crime analysts to locate ongoing series, to assign correct attribution for past crimes, and to prevent further crime. There is convincing evidence that time and resources spent on place-based approaches, where police target specific locations and times, result in crime reduction and safer neighborhoods [Weisburd2012, Weisburd and Mazerolle2000,Sherman, Garlin, and Buerger1989]. Research on crime pattern detection can lead directly to place-based approaches, and any advance in this area could lead directly to reductions in crime. (In contrast, it is well known that random preventive patrol does not deter crime [Sherman and Eck2002,Kelling et al.1974].)

Despite its critical importance for public safety, there are few academic works on this problem (e.g., [Dahbur and Muscarello2003, Nath2006, Brown and Hagen2003, Lin and Brown2003]). Most predictive policing software has the capability to detect general background levels of crime density, which are much easier to predict than specific patterns of crime. Detailed information about each crime is not necessary to predict background levels, whereas it is necessarily required to determine the M.O. However, knowing the M.O. of a crime series can be actionable, whereas estimation of background levels may not be directly actionable. For instance, if suspect information is available for at least one crime in a series, then this suspect information can be linked to all crimes in the series. Further, if a current crime series has predictable times and locations (for instance, a pickpocket targeting a single cafe at regular intervals throughout the week), actions can be taken to stop the pattern. Also, if a current crime series has the same M.O. as a past crime series for which the offender is known, then the suspect from the older series could be a potential offender for the current crime series.

Without the capability of automatically detecting specific series of crime, it is possible that crime series may take much longer to identify, or may never be identified. This is especially problematic for certain types of crime - for instance for housebreaks (burglaries) there is

often no suspect information, as the crimes take place when residents are not present. Housebreaks can be extremely difficult crimes to solve, and nationwide only 14% of housebreaks are solved [Weisel2002]. In this paper, we aim directly at identifying series in housebreaks in an automated way.

Crime series detection can be viewed as a type of clustering problem where the M.O. defines the set of relevant features for each cluster. We cannot determine in advance what the exact M.O. of an undetected crime series will be. Generally speaking, we need to reveal groups of objects that are similar on an unknown subset of their features. We say that such sets exhibit a *pattern-specific* similarity. The pattern-specific similarity cannot be learned since it may be true for only a small number of crimes (perhaps less than 10).

Though we cannot characterize exact M.O.'s before we see them, we can characterize the type of M.O.'s we expect generally - for instance, crimes in a series are often close in time and space. We define a *pattern-general* similarity that encodes which factors are generally common to most crime series. It is learned from past crime series; for instance, time and space have high pattern-general weights. The pattern-general similarity induces a *similarity graph* over the set of crimes. We use this graph to examine whether sets of crimes are generally similar to historical crime series.

The main hypothesis in this work is that most crime patterns have a *core* of crimes that exemplify the M.O. of the series. If we can locate all small cores of crime, we should thus have located parts of most crime series'. This hypothesis is based on the intuition of analysts, and has the dual purpose of assisting with computation - we can indeed consider all possible small subsets to calculate whether they are plausible core sets. The core sets are found using an integer linear programming (ILP) formulation, which specifies that core sets must have pattern-specific and pattern-general similarity. Once the core sets are found, we construct the full crime series by merging overlapping core sets. We prove that merging core sets preserves desired properties.

This method was tested on the full housebreak database from the Cambridge Police Department containing information from thousands of crimes from over a decade. It was able to provide new insights into true patterns of crime committed in Cambridge.

Related Work The problem we consider is a clustering problem, but where the similarity measure between objects is supervised. Our work relates to various subfields of clustering (see for instance [Kriegel, Kröger, and Zimek2009]), including pattern-based clustering [Wang et al.2002, Pei et al.2003] which is a semi-supervised approach (unlike ours - we do not use test data at training time), subspace clustering (e.g., [Domeniconi et al.2004, Vidal2011]) which detects all clusters in all subspaces, and work at the intersection of dense subgraph mining and pattern mining in feature graphs that is somewhat similar to ours [Moser et al.2009, Günnemann, Boden, and Seidl2011].

Other work on space-time event detection is relevant [Neill and Cooper2010], though the goal is to detect patterns where the frequency of records is higher than an expected frequency, whereas here we cannot accurately estimate the expected frequency due to high dimensionality and sparseness.

Most previous work on crime series detection [Nath2006, Lin and Brown2003] have pattern-general weights that come from experts (like our baselines) and are not learned from data (with [Brown and Hagen2003] as an exception), and do not consider pattern-specific aspects, and thus cannot capture specific M.O.'s of crimes. Some past approaches have flaws that require heuristic post-processing to handle [Dahbur and Muscarello2003]. Our previous attempt at solving this was an iterative approach that did not consider core sets [Wang et al.2013].

Main Contributions Our clustering method is novel in that it (i) uses both "pattern-general" and "pattern-specific" aspects of patterns, (ii) the pattern-general similarity between nodes is learned from past clusters, allowing us to use graph-based pattern-mining methods and (iii) the pattern specific aspects require dense subgraph mining in subspaces, and merging of these subgraphs. The method was developed to handle the specialized aspects of crime series data (including pattern-specific M.O.'s).

Model formulation

Our data consists of entities (crimes) $\mathcal{V}$ each of which has a vector of features. Define D_j as the set of possible values for feature j, $j = 1...J$. In our database specifically we have features including time, space, whether the crime occurred on a weekday, location of entry (front door, back door, ground window, etc.), means of entry (shoved, pried, forced, etc.), type of premise (apartment, single-family house, etc.) an indicator variable for 'ransacked', suspect information (which is rarely present), and so on. The model we introduce below is general and can be applied to problems beyond crime data mining; however, we use terminology specific to our problem in our exposition.

Define s_j to be a symmetric similarity function on the jth feature $s_j : D_j \times D_j \rightarrow [0, 1]$, where $s_j(v_l, v_k)$ measures the similarity between crimes v_ℓ and v_k in jth feature. In our case most features are categorical, some are ranges, and some are spatial coordinates. The similarity measures are discussed in [Wang et al.2013] (and are thus not discussed here due to space limitations). The average similarity of a set of crimes in feature j is defined as the cohesion of the set:

Definition 1. *(**Cohesion**$_j$) For a set of crimes V, the cohesion in the jth feature is the mean of pairwise similarities, $Cohesion_j(V) = \frac{1}{|V|(|V|-1)} \sum_{v_l, v_k \in V} s_j(v_l, v_k)$.*

The defining features of a pattern are those with sufficiently high cohesion.

Definition 2. *(**Defining feature**) Defining features for V are those that satisfy $Cohesion_j(V) \geq \theta_j$. The set of defining features for set V is denoted by $\Lambda(V)$.*

The defining features characterize the M.O. of the crime series. If several housebreaks happen in the same neighborhood, around the same time of day, within the same month, and the location of entry is always a window, regardless of the differences in other features, these similarities would indicate that the crimes could have been committed by the same offender. The features "geographic location," "time of day," "time between crimes," and "location of entry" characterize the M.O. for this particular crime series. When we later define core sets, the pattern-specific statistic of interest is the number of defining features of V.

Let us switch from pattern-specific definitions to pattern-general definitions. We would like our core sets to have crimes that are similar to each other in the pattern-general sense. This is because, for instance, patterns that are spread very far apart in time and space are very unlikely to be a pattern, and pattern-general similarity allows us to weight important features like time and space more highly. We will learn a set of weights $[\lambda_1, ..., \lambda_j, ...\lambda_J]$ from past crime data that will provide the importance of each feature within a linear combination. We will search for core sets that have high pattern-general cohesion, defined as follows:

Definition 3. *(**Pattern-general cohesion**) The pattern-general cohesion of V is the weighted sum of cohesions over the features: $\sum_{j=1}^{J} \lambda_j Cohesion_j(V)$.*

Having high pattern-general cohesion is not sufficient; for V to be a core of a series, we require additional constraints to ensure the core is connected in the graph theoretic sense (and thus is not composed of two separate clusters for instance), and that it has sufficient connectivity. To define these constraints we thus need to define the similarity graph. To construct edges for the graph, we first define a metric to measure the similarity between two crimes, as follows:

Definition 4. *(**Pattern-general similarity**) The pattern-general similarity is a weighted sum of similarity measures for each feature, using the pattern-general weights $[\lambda_1, \lambda_2, ...\lambda_J]$.*

$$\gamma(v_\ell, v_k) = \sum_{j=1}^{J} \lambda_j s_j(v_\ell, v_k). \quad (1)$$

We define the similarity graph to contain edges between crimes that are sufficiently close in the pattern-general sense.

Definition 5. *(**Similarity graph**) A similarity graph is an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, v_2, ...v_n\}$ is the node set, $\mathcal{E}$ is the edge set, $\mathcal{E} = \{\{v_\ell, v_k\} | \gamma(v_\ell, v_k) \geq \Delta, v_\ell, v_k \in V, v_\ell \neq v_k\}$.*

We learn both the weights $\{\lambda_j\}_j$ and cut-off threshold Δ in Section . The choice of definition of the similarity graph can be changed to any other measure, and the procedure for finding core sets below can be directly followed, no matter what the definition of the pattern-general similarity is. For instance, we could eliminate the pattern general similarity all together by setting the threshold Δ to be very low, and that way we look only for pattern-specific similarity.

Imposing a graph structure on the crimes eases computation, in the sense that we are now only looking for connected sets of the graph. Note that we cannot simply maximize the number of defining features and/or the pattern-general cohesion over all subsets of crime, as it would favor choosing very small sets of crime. To alleviate this problem, we specify the size of the core set $|V|$ we are looking for, and the number of defining features, and maximize the pattern-general cohesion. Once we find the core sets, we merge or expand them to find the rest of the pattern. We call patterns formed by merging other patterns together *series-like* patterns.

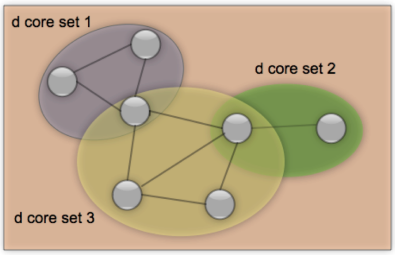

Figure 1: A d series-like pattern consisting of three d core sets of various sizes

Core sets

A *d-core set* is a set of crimes that exhibit similarity in a feature subspace of d dimensions. A d core set has d defining features that are not predetermined. Further, crimes in a d-core set need to be well connected in the similarity graph.

Definition 6. *(**d-core set**) A similarity graph $G = (V, E)$ with density threshold α is called a d-core set if it satisfies the core set constraints:*

- *Pattern specific constraint: the size of the defining feature set of the graph is equal to d, $|\Lambda(G)| = d$.*
- *Pattern general constraints: G is connected, and G is dense, $\frac{|E|}{|V|(|V|-1)} \geq \alpha$. That is, the fraction of possible edges in the graph exceeds α.*

Two parameters, d and α, control the property of the core set in a pattern-specific and pattern-general way, respectively.

The pattern-general constraints should be thought of as much looser than the pattern-specific constraints, as we will include many unnecessary edges in the similarity graph. For the set of crimes to be feasible in the pattern-specific sense is much more difficult, as crimes in the pattern need to be similar to each other on average in d separate ways.

We find core sets $G = (V, E)$ using an optimization method to maximize the pattern-general cohesion while satisfying the core set constraints.

$$\begin{aligned} &\underset{G}{\text{maximize}} && \sum_{j \in J} \lambda_j \text{Cohesion}_j(G) \\ &\text{subject to} && |V| = n \\ &\text{Core set constraints:} && \begin{cases} |\Lambda(G)| = d, \\ G \text{ is connected}, \\ \frac{|E|}{|V|(|V|-1)} \geq \alpha. \end{cases} \end{aligned} \tag{2}$$

We propose a binary integer linear formulation for the optimization problem (2), allowing us to solve it with mathematical programming technology (CPLEX). Because the formulation is linear, we benefit from the latest technology in solvers, which have improved exponentially over the last decade. We also benefit from a guarantee on the optimality of the solution. Let $m = |\mathcal{V}|$, the number of crimes in the database. Let n be the size of the pattern we want to discover, and we loop through possible values of n. We define an $m \times m$ similarity matrix for each feature $\mathbf{S}_1, \ldots, \mathbf{S}_J$ with elements $\mathbf{S}_j(\ell, k) = s_j(v_\ell, v_k)$. Let $\mathbf{X}$ be the matrix of binary decision variables defining the core set. $\mathbf{X}(\ell, k)$ is 1 if a pair of crimes ℓ and k are in core set G, which is the same as $\mathbf{X}(k, \ell)$, so matrix $\mathbf{X}$ is symmetric. On the diagonal, $\mathbf{X}(\ell, \ell)$ represents whether crime ℓ is in core set G. We use $d_1, \ldots, d_J \in \{0, 1\}$ to indicate whether feature j is a defining feature, or equivalently, whether $\text{Cohesion}_j(G) \geq \theta_j$. Let $\mathbf{E}$ be the adjacency matrix for the (pattern-general) similarity graph, where $\mathbf{E}(\ell, k) = 1$ if $\{v_\ell, v_k\} \in E$, and $\mathbf{E}(\ell, k) = 0$ otherwise. Since the graph is undirected, $\mathbf{E}$ is symmetric. We set $\mathbf{E}(\ell, \ell) = 1$ for computational simplicity. $(\mathbf{E} \circ \mathbf{X})$ is the Hadamard product of matrix $\mathbf{E}$ and $\mathbf{X}$, where $(\mathbf{E} \circ \mathbf{X})(\ell, k) = \mathbf{E}(\ell, k) \cdot \mathbf{X}(\ell, k)$. M is a large auxiliary parameter for formulating the problem with a big-M formulation, and ϵ is small. With this notation, the optimization problem (2) can be reformulated:

$$\max_{\mathbf{X}, \{\mathbf{d_j}\}_\mathbf{j}} \quad \sum_j \lambda_j \sum_{\ell,k} (\mathbf{S}_j \circ \mathbf{X})(\ell, k) \text{ s.t.}$$

$$\sum_\ell \mathbf{X}(\ell, \ell) = n \tag{3}$$

$$\frac{1}{n(n-1)} \sum_{\ell,k} (\mathbf{S}_j \circ \mathbf{X})(\ell, k) - M d_j \leq \theta_j - \epsilon \quad \forall j \tag{4}$$

$$\frac{1}{n(n-1)} \sum_{\ell,k} (\mathbf{S}_j \circ \mathbf{X})(\ell, k) - M d_j \geq \theta_j - M \quad \forall j \tag{5}$$

$$\sum_{j \in J} d_j = d \tag{6}$$

$$\mathbf{X}(\ell, k) = \mathbf{X}(k, \ell) \quad \forall \ell, k \tag{7}$$

$$\mathbf{X}(\ell, k) \leq \mathbf{X}(k, k) \quad \forall \ell, k \tag{8}$$

$$\mathbf{X}(\ell, \ell) + \mathbf{X}(k, k) \leq \mathbf{X}(\ell, k) + 1 \; \forall \ell, k \tag{9}$$

$$(\hat{\mathbf{E}}^{n-1} \circ \mathbf{X})(\ell, k) \geq \mathbf{X}(\ell, k) \quad \forall \ell, k \tag{10}$$

$$\mathbf{X}(\ell, k), d_j \in \{0, 1\} \quad \forall \ell, k, j. \tag{11}$$

Let us derive the objective. Since $\mathbf{X}(\ell, k) = 1$ if and only if both crimes ℓ and k are in the core (that is they are in graph G that we discover), we have the following:

$$\frac{1}{n(n-1)} \sum_{\ell,k} (\mathbf{S}_j \circ \mathbf{X})(\ell, k) = \text{Cohesion}_j(G). \tag{12}$$

The objective is the pattern-general cohesion. Equation (3) ensures that the core sets we discover are of size n. Constraints (4), (5) and (6) are the pattern-specific constraints. Constraint (4) forces $d_j = 1$ when $\text{Cohesion}_j(G) \geq \theta_j$, where the strict inequality is enforced by ϵ. Constraint (5) forces $d_j = 0$ when $\text{Cohesion}_j(G) < \theta_j$. Constraint (6) specifies the number of defining features as d. The symmetry of $\mathbf{X}$ is enforced by (7). Constraints (8) and (9) imply $\mathbf{X}(\ell, k) = 1$ iff both $\mathbf{X}(\ell, \ell)$ and $\mathbf{X}(k, k)$ are 1 and 0 otherwise.

Expression (10) is a pattern-general constraint. Our formulation does not enforce the core set to be connected, but it does enforce something weaker, namely that each node in a core set of size n is at most $n - 1$ steps along the similarity graph from any other node in the core set. We handle the connectivity afterwards, by examining the result to ensure that it is connected, and labeling it as infeasible if not. To handle the constraint that each node in the core set is at most distance $n-1$ from every other node, we recall that in graph theory, if node v_k is reachable from node v_ℓ in exactly q steps, $\mathbf{E}^q(k, \ell) > 0$. If node v_k is reachable from node v_ℓ in at most $n - 1$ steps, it means that at least one of $\mathbf{E}^q(k, \ell) > 0$ for $q \leq n - 1$. We define the following matrix $\hat{\mathbf{E}}^{n-1}$, where an element $\hat{\mathbf{E}}^{n-1}(k, \ell)$ indicates if node k and node ℓ are distance at most $n-1$ steps along the graph.

$$\hat{\mathbf{E}}^{n-1}(\ell, k) = \begin{cases} 1 & (\mathbf{E} + \mathbf{E}^2 + \cdots + \mathbf{E}^{n-1})(\ell, k) > 0 \\ 0 & \text{otherwise.} \end{cases}$$

Thus, (10) forces that if v_ℓ and v_k are both in the pattern, they must be at most distance $n - 1$ along the graph.

The pattern-general density constraint is handled similarly to the connectivity constraint, where feasibility is checked for each solution, and infeasible solutions are removed.

The integer program finds one solution at a time. In order to avoid finding a solution that was found previously, we introduce a constraint for each previous solution found. Suppose in the t-th run, the crimes in the solution are $\mathcal{Q}_t$, i.e. $\mathbf{X}(k, k) = 1$ if $k \in \mathcal{Q}_t$, $\mathbf{X}(k, k) = 0$ otherwise. The constraint we add before running the $t + 1$-th time is

$$\sum_{k \in \mathcal{Q}_t} \mathbf{X}(k, k) \leq n - 1. \tag{13}$$

This constraint will exclude the current solution from the feasible region and we will obtain a different solution in the next run if any feasible solutions remain.

Since all of the matrices are symmetric, in practice we keep only the upper (or lower) triangle, including the diagonal, to compute all of the sums.

Merging core sets

By our main hypothesis, the vast majority of crime series contain a core set. This means that by finding all core sets, we would have located the vast majority of all crime series'. We now grow the rest of the crime series' from the core sets by merging them together. One advantage of merging is that it allows the pattern to dynamically change, as the defining features from the merged core sets are not always equal to each other. Consider a burglar's M.O. with a shifting means of entry. At first he enters through unlocked doors, then he starts to use bodily force to open doors, and later he learns to use a screwdriver to pry the door open. His full pattern thus consists of several smaller d-core sets. This suggests a more flexible definition of pattern than a simple core set. We provide such a definition below.

Definition 7. *(d-**series-like pattern**) A graph $G = (V, E)$ is called a d-series-like pattern with defining feature set $\Pi(G)$ of size d if it satisfies:*

- *Pattern general constraint: G is connected.*
- *Pattern specific constraint: Each node u in G is contained in at least one subgraph $G' \subseteq G$ that is a d'-core set with defining features that include set $\Pi(G)$, that is $\Pi(G) \subseteq \Lambda(G')$, $d \leq d'$.*

Note that if a graph is a d-series-like pattern, it is also a $(d-1)$-series-like pattern, a $(d-2)$-series-like pattern, and so on. The pattern specific constraints for d-series-like patterns are looser than those for d-core sets. A d-series-like pattern may not be a d-core set. On the other hand, a d-core set is a special case of a d-series-like pattern. Before we proceed, we must ensure that merging is justified.

Theorem 1. *(**The set of series-like patterns is closed under merging.**) Suppose G_1 is a d_1-series-like pattern and G_2 is a d_2-series-like pattern. If $G_1 \cap G_2 \neq \emptyset$, then $\hat{G} = G_1 \cup G_2$ is a d-series-like pattern, with defining features $\Pi(G_1) \cap \Pi(G_2)$, $d = |\Pi(G_1) \cap \Pi(G_2)|$.*

Proof. First, since G_1 and G_2 are connected and $G_1 \cap G_2 \neq \emptyset$, the union of them is also connected, i.e, $\hat{G}$ satisfies the pattern general constraint. then for $\forall$ nodes $u \in G_1$, $\exists$ a d_1-core set G_u^1 such that $u \in G_u^1$ and $\Pi(G_1) \cap \Pi(G_2) \subseteq \Pi(G_1) \subseteq \Lambda(G_u^1)$, and for $\forall$ nodes $u \in G_2$, $\exists$ a d_2-core set G_u^2 such that $u \in G_u^2$ and $\Pi(G_1) \cap \Pi(G_2) \subseteq \Pi(G_2) \subseteq \Lambda(G_u^2)$. So, either way, the defining feature set includes $\Pi(G_1) \cap \Pi(G_2)$. This means that for any node $\hat{u} \in \hat{G}$, $\exists$ a core set $\hat{G}_u$ such that $\Pi(G_1) \cap \Pi(G_2) \subseteq \Lambda(\hat{G}_u)$. □

This leads directly to the following:

Corollary 1. *Suppose G_1 is a d_1-series-like pattern, G_2 is a d_2-series-like pattern,..., G_n is a d_n-series-like pattern, and $G_1 \cup ... \cup G_n$ is connected, and*

$$|\Pi(G_1) \cap \Pi(G_2) \cap ... \cap \Pi(G_n)| = d.$$

Then $G_1 \cup ... \cup G_n$ is a d-series-like pattern.

These properties lead to the following breadth first search algorithm for mining d-series-like patterns. We start from the set of core sets that we found using the integer program. These core sets are *candidates* for merging. We also maintain an *active pattern set* that contains the d-series-like patterns that we are not done constructing. We keep the d-series-like patterns that we are done constructing in a *maximal pattern set*. To start, the active pattern set contains all of the d-core sets we found. For each active pattern, we iterate through the candidates to see if they meet the merging criteria provided just below. If a merge is possible, and if the merged set had not already been previously created, we append the merged pattern to the active pattern, and continue iterating through the candidates. If there are no candidates that can be merged with the active pattern at all, then the active pattern is maximal, and it is placed in the maximal pattern list. The *merging criteria* for pattern G_1 and G_2 to get a d-series-like pattern $\hat{G}$ is

- $G_1 \cap G_2 \neq \emptyset$
- $|\Pi(G_1) \cap \Pi(G_1)| \geq d$.

The merging algorithm is formulated below:

Algorithm 1 Merging core sets

```
INPUT: d, core sets, each with ≥ d defining features
candidate list ← core sets
active set ← core sets, each with defining features
maximal set ← ∅
while active set ≠ ∅ do
  G_current ← any element in active set
  Π_current ← defining features from G_current
  isMaximal ← TRUE;
  for all G_j ∈ candidates, G_j ⊄ G_current do
    if G_current ∩ G_j ≠ ∅, |Π(G_current) ∩ Λ(G_j)| ≥ d
    then
      Ĝ ← G_current ∪ G_j
      Π(Ĝ) ← Π(G_current) ∩ Λ(G_j)
      if Ĝ does not exist in active set or maximal
      set then
        isMaximal ← FALSE;
        append Ĝ to active set
      end if
    end if
  end for
  if isMaximal==TRUE then
    remove G_current from active set, put into maxi-
    mal set
  end if
end while
OUTPUT: maximal set
```

Learning the similarity graph

Recall that the similarity graph is constructed by connecting pairs of nodes whose pattern general cohesion

is above a threshold Δ. The pattern general cohesion γ is defined in (1) as a weighted sum of pairwise similarities in different features, with pattern-general weights $\boldsymbol{\lambda} \in \mathcal{R}^J$. The set of coefficients $\boldsymbol{\lambda}$ and Δ are parameters that we learn from data.

We have 51 historical crime series' that have been identified as true crime series by crime analysts that we used to train the model. The idea is to optimize over the historical training patterns that they are as close as possible to being connected subgraphs according to the pattern-general similarity. This means we want each crime in a historical pattern to be close to at least one other crime in the same pattern. At the same time, we want crimes within a historical pattern to be distant from crimes not in the same pattern.

We will encourage each crime within a pattern to be close to at least one other crime in the pattern, but with some slack; we want the graph to be conservatively constructed and include more edges than needed, and to allow the pattern-specific terms to provide most of the accuracy, as discussed earlier. The condition for crime ℓ to be close to at least one other crime within its pattern (with some slack) is:

$$\max_{\{k:k\in\text{pattern}(\ell)\}} \sum_{j=1}^{J} \lambda_j s_j(\ell,k) \geq \Delta - \epsilon_\ell. \qquad (14)$$

Conversely, crimes that do not belong to the same pattern should not be very similar:

$$\sum_{j=1}^{J} \lambda_j s_j(\ell,k) \leq \Delta + \xi_{\ell k}, \qquad (15)$$

true for all ℓ and k s.t. $k \notin \text{pattern}(\ell)$. Our goal is to minimize the total weighted slack. Loosely, we compute:

$$\min_{\boldsymbol{\lambda},\Delta} \sum_{\substack{\text{crimes in the}\\ \text{same pattern}}} \text{slack} + C_1 \sum_{\substack{\text{crimes not in the}\\ \text{same pattern}}} \text{slack} + C_0\|\boldsymbol{\lambda}\|_0,$$

where $\|\boldsymbol{\lambda}\|_0$ is the ℓ_0 semi-norm of $\boldsymbol{\lambda}$, which encourages sparsity in $\boldsymbol{\lambda}$. The constant C_1 is set very small, so we make sure that all edges that should be present are present, and consider removal of unnecessary edges as a secondary goal.

We propose a mixed-integer programming (MIP) formulation for solving this. In what follows, decision variables $y_{\ell k}$ are binary, and they select a crime k that is most similar to a given crime ℓ within the same pattern. That is, they encode the max from constraint (14). The formulation is:

$$\min_{\boldsymbol{\lambda},\Delta,\{y_{\ell,k}\}_{\ell,k},\{\beta_j\}_j} \sum_{\ell\in\{1,2\ldots m\}} \epsilon_\ell + C_1 \sum_{\substack{\ell\in\{1,2\ldots m\}\\ \{k:k\notin\text{pattern}(\ell)\}}} \xi_{\ell k} + C_0 \sum_{j=1}^{J} \beta_j$$

such that

$$\sum_{j=1}^{J} \lambda_j s_j(\ell,k) \geq (\Delta - \epsilon_\ell) + M(y_{\ell k} - 1), \quad \forall \ell,\ \forall k \text{ s.t. } k \in \text{pattern}(\ell) \qquad (16)$$

$$\sum_{k:\ k\in\text{pattern}(\ell)} y_{\ell k} = 1 \qquad \forall \ell\ (17)$$

$$y_{\ell k} \in \{0,1\}, \qquad \forall \ell, k \qquad (18)$$

$$\sum_{j=1}^{J} \lambda_j s_j(\ell,k) \leq \Delta + \xi_{\ell k}, \quad \forall \ell,\ \forall k \text{ s.t. } k \notin \text{pattern}(\ell) \qquad (19)$$

$$\sum_{j=1}^{J} \lambda_j = 1, \qquad (20)$$

$$\lambda_j \geq 0 \qquad \forall j\ (21)$$

$$\lambda_j \leq \beta_j \qquad \forall j\ (22)$$

$$\beta_j \in \{0,1\} \qquad \forall j. (23)$$

Constraint (16) comes from (14). It forces $y_{\ell,k}$ to be chosen correctly so that if k is the crime closest to ℓ, then $y_{\ell,k}$ will be 1. This is because ϵ_ℓ is minimized within the objective, so $y_{\ell,k}$ is necessarily going to correspond to the index where ϵ_ℓ is minimized, and where the similarity is maximized within (16). Constraints (17) and (18) further define $y_{\ell,k}$'s by stating that a crime in the pattern needs only to be connected to one closest neighbor ℓ in the pattern (this is the requirement of connectivity), and its entries are binary. Constraint (19) comes from (15). The value $\sum_j \beta_j$ is the ℓ_0 norm of $\boldsymbol{\lambda}$. The β_j's are decision variables where if $\beta_j = 1$, λ_j is non-zero. This formulation is not difficult on modern MIP solvers because it is linear, and there is a guarantee on the optimality of the solution.

Performance

Our data set was provided by the Crime Analysis Unit of the Cambridge Police Department in MA, USA. It has 7,067 housebreaks that happened in Cambridge between 1997 and 2011, containing 51 hand-curated patterns contained within the 4,864 crimes between 1997 and 2006. (Patterns from 2007 to 2012 were not assembled at the time of writing.) Crime attributes include geographic location, date, day of 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. Data were processed using the similarity functions $\{s_j\}_j$ discussed by [Wang et al.2013] where each pairwise feature is mapped into a number in $[0,1]$. These similarity measures consider the baseline frequency of each possible outcome for the categorical variables. We took the 51 hand-curated patterns, and divided them randomly into four subsets (folds) with sizes 12 or 13 patterns each. We used 3 of the 4 folds to learn the pattern general weights and tested on the remaining fold for the experiments discussed below.

Baselines As this problem is fundamentally a clustering problem, we compare with several varieties of hierarchical agglomerative clustering and incremental nearest neighbor approaches. For these baselines, we use

several different schemes to iteratively add discovered crimes, starting from pairs of nodes with high similarity γ, which is a weighted sum of the attribute similarities:

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

Unlike our method where the weights are learned, the weights $\hat{\boldsymbol{\lambda}}$ for the baselines were provided by crime analysts based on domain expertise, similar to [Nath2006, Lin and Brown2003].

Hierarchical agglomerative clustering (HAC) begins with each crime as a singleton cluster, and iteratively merges the clusters based on the similarity measure between clusters. *Nearest neighbor classification* (NN) first selects pairs of crimes with high similarity and then iteratively grows a cluster by adding the nearest neighbor crime to the cluster.

HAC and NN were used with three different criteria for cluster-cluster or cluster-crime similarity: *Single Linkage* (SL), which considers the most similar pair of crimes, *Complete Linkage* (CL), which considers the most dissimilar pair of crimes, and *Group Average* (GA), which uses the averaged pairwise similarity [Hastie et al.2005]. When the nearest neighbor algorithm is used with the S_{GA} measure defined below with weights provided by crime analysts, it is similar to the Bayesian Sets algorithm and how it is used for set expansion [Ghahramani and Heller2005, Letham, Rudin, and Heller2013].

$$\begin{aligned} S_{SL}(G_1, G_2) &:= \max_{v_k \in G_1, v_\ell \in G_2} \gamma(v_k, v_\ell) \\ S_{CL}(G_1, G_2) &:= \min_{v_k \in G_1, v_\ell \in G_2} \gamma(v_k, v_\ell) \\ S_{GA}(G_1, G_2) &:= \frac{1}{|G_1||G_2|} \sum_{v_k \in G_1} \sum_{v_\ell \in G_2} \gamma(v_k, v_\ell). \end{aligned} \tag{24}$$

Evaluation metrics There are two levels of performance we evaluate - *pattern-level* and *object-level*.

Pattern level precision and recall
We evaluate the quality of the core set detector using "pattern-level" precision and recall. The d-core sets tend to be small for reasonable large d. They are used for discovering larger patterns, so we evaluate their accuracy of detecting real patterns; if a real pattern is missed completely by our core set detector, there is no way for us to recover from this in order to detect it. If a core set covers more than one pattern, this is also a bad seed for further mining, since it would generate misleading defining features that do not characterize any real patterns. Thus, we call core sets that cover one and only one real pattern *good* core sets. The pattern level precision and recall are both defined using good core sets. N denotes the number of core sets we discover.

$$\text{P-Precision (core sets)} = \frac{\sum_i^N \mathbf{1}(\text{core set } i \text{ is } \textit{good})}{N} \tag{25}$$

$$\text{P-Recall (core sets)} = \frac{\sum_i^N \mathbf{1}(\text{core set } i \text{ is } \textit{good})}{|\mathcal{P}|}. \tag{26}$$

Note that pattern level precision should be large, as many of the core sets we discover should apply to the same patterns (inflating the reported precision values).

Object-level precision and recall
We evaluate the full pipeline for generating series-like patterns using "object-level" precision and recall. To do this, for each pattern discovered, we determine how close it is to one of the real patterns. If the discovered pattern overlaps only one real pattern, then we call this the *dominating pattern* and evaluate precision and recall with respect to crimes in that pattern. If the series-like pattern overlaps more than one real pattern, we assign the dominating pattern to be the real pattern possessing the most crimes that overlap with our discovered pattern. Note that it is possible for the recall not to grow with the size of the discovered pattern, as the dominating real pattern could change as the discovered pattern grows larger. The definitions of object-level precision and recall for a d-series-like pattern $G = (V, E)$ are as follows:

$$\text{O-Precision(G)} = \frac{\sum_{\ell=1}^{|V|} \mathbf{1}(\ell \in \text{dominating pattern})}{|V|} \tag{27}$$

$$\text{O-Recall(G)} = \frac{\sum_{\ell=1}^{|V|} \mathbf{1}(\ell \in \text{dominating pattern})}{|V_{\text{dominating pattern}}|} \tag{28}$$

where $|V_{\text{dominating pattern}}|$ is the number of crimes in the dominating real pattern.

Computational gain from similarity graph The first step in our method is to learn the similarity graph. The similarity graph provides a computational gain in that it creates constraints on possible core sets, reducing the feasibility region of the ILP. For this similarity graph, recall that we desire crimes in the same real pattern to be connected to each other. We call edges connecting crimes that belong to the same pattern *good edges*. If we have constructed the similarity graph well, the similarity graph should have a higher percentage of good edges than if we had simply used the full graph consisting of all possible edges. If we remove a few good edges in the process, this is not problematic as long as the true patterns are still connected in the similarity graph - this will be assessed when we assess the quality of the core sets and the full pipeline next.

Table 1 shows the percentages of good edges in both the similarity graph and the full graph, and one can

see that the learning we did generally tends to reduce the number of unnecessary edges by a factor of 7 or 8 in each of the test folds. This reduction substantially reduces computation for the core set finder.

Table 1: Test results of weights learning algorithm

	Good edges% in similarity graphs	Good edges % in complete graphs	Training time
1	1.84	0.26	3.64×10^3 s
2	2.42	0.33	9.58×10^3 s
3	3.34	0.42	6.20×10^3 s
4	2.95	0.34	5.41×10^3 s

Pattern-general weights The pattern-general weights come from the learning step for the similarity graph. In Figure 2 we report the mean over the test folds of the pattern-general weights we discovered. The highest weights are similarity in distance, number of days apart, suspect information, whether residents are present, and means of entry (e.g., pried, forced, cut screen).

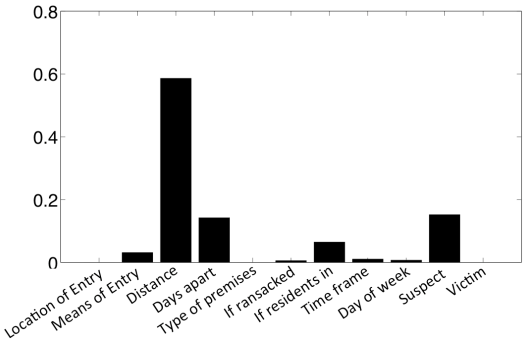

Figure 2: Pattern general weights

Time windowing Consider solving the ILP for finding core sets on data from 4,864 housebreaks. Note that if we were to search for patterns of size 10 among 1,000 crimes, this would mean investigating $\binom{1000}{10} \approx 2.634 \times 10^{23}$ possible subsets. Further, CPLEX would need to handle $1{,}000^2$ constraints (8) and (9) of the optimization problem. Luckily it is unlikely that a crime series would possess a core of size 10, but still we need to find ways to reduce computation.

Because the pattern-general weight on closeness in time is so high, we determined that we would be unlikely to miss true core sets if we considered windows of time that include at least 200 crimes. We thus indexed the housebreak records in chronological order and created overlapping windowed blocks of 200 crimes each, where neighboring blocks have an overlap of 100 crimes. Therefore we solve ILPs among crime subsets $\{1, \cdots, 200\}$, $\{100, \cdots, 300\}$, $\cdots$, $\{4700, \cdots, 4864\}$. In each subset, we input the number of defining features d and core set size n, and then iteratively run the ILP to get all feasible solutions, by adding the constraint (13) after each iteration to avoid returning repeated solutions.

Evaluation of mining core sets

We chose performance evaluation metrics from information retrieval, and for some of these metrics, we need to rank the discovered core sets by a scoring function. This scoring function represents how certain we are that these core sets are real. We use a scoring function that is a weighted version of pattern-general cohesion and the (pattern-specific) number of defining features, as we desire core sets that are both tight in the pattern-general sense and in the pattern-specific sense. Here is the score function:

$$\text{Score}(G) = \sum_{j=1}^{J} \lambda_j \text{Cohesion}_j(G) + \frac{1}{6} \cdot d. \qquad (29)$$

We ordered the discovered core sets in decreasing order of the scores. Note that the first evaluation below does not require these scores, but the second and third do.

1) Core Sets with different d We expect that discovered patterns with more defining features are more likely to be true crime series. Figure 3 shows how the precision increases with the number of defining features d. In this figure we consider only core sets of the same size (3 crimes). There are no overlapping core sets between the three bars, as each core set is used once with its exact number of defining features d, which is 6, 7, or 8. The number of discovered core sets for $d = 6$ is 1072, $d = 7$ is 215 and for $d = 8$ is 36. There were too few core sets with more than 8 defining features to reliably calculate precision.

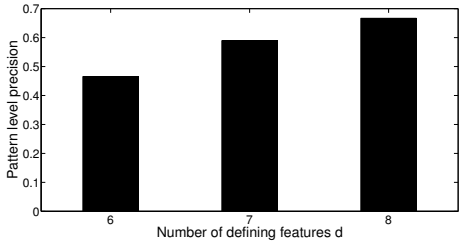

Figure 3: Pattern level average precision of d core sets of different sizes.

2) Core Sets with different size n We use the scoring function (29) as a filter to pick the best 1,000 core sets, from each of sizes 3, 4 and 5 and discarded the other discovered score sets. Larger core sets have higher chances of hitting a pattern, since there are more crimes in the core set; however, they also have higher chances of hitting more than one real pattern. As shown in

Figure 4, core sets of size 4 have a much higher pattern-level precision than core sets of size 3, but core sets of size 5 do not have noticeable gains over size 4. This is because the increased probability of hitting a real pattern cancels with the increased probability of hitting more than one real pattern.

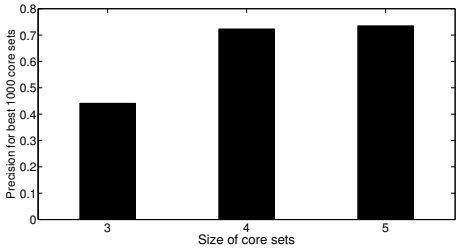

Figure 4: Pattern-level precision of core sets with different sizes

3) P-Precision - P-Recall curve We generated a full list of core sets of size 3 with d between 6 and 8, and ranked the core sets according to their scores. As we moved down the list, we evaluated pattern-level precision and recall at each step. We also did the same procedure with the baseline iterative nearest neighbor method used for generating core sets of size 3, using all of the similarity measures in (24). (Note that for HAC we cannot control the size of cores for evaluation.) The precision-recall curves for all four methods averaged over the test folds are plotted in Figure 5. It is clear that our core set finder is substantially better than the baselines, though that is not surprising given that it searches globally for the best core sets.

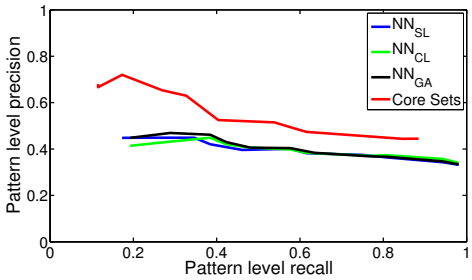

Figure 5: Avg pattern-level precision vs. recall

Evaluation for mining series-like patterns

We evaluated the quality of our full pipeline and the baseline methods as follows. For our method, the results of the core set finder are merged using the merging algorithm. The merging process stops when there are no two patterns satisfying the merging criteria. After all the series-like patterns were discovered, we evaluated the average object-level precision and recall for all the patterns and over all the test folds, plotted as a point on Figure 6. For HAC, we simply iterated it, stopping at a threshold where recall was approximately equivalent to our method, and again reported average object-level precision and recall on Figure 6. For the nearest neighbor method, after each element was added to a growing pattern, we evaluated precision and recall to trace out a precision-recall curve. All three metrics in (24) were used for HAC and nearest neighbors. We note that for the same level of recall, the precision attained by our method was quite a bit higher than that of other methods.

There is still a lot of room for improvement, as with precision on the order of 53%, we capture approximately 18% of the crimes identified by analysts. This might be improved by making the core set finder less conservative, finding a way to incorporate crimes that are not in cores, or possibly by working with domain experts to improve the database used for evaluation. We note that the baseline methods work surprisingly well, and are reasonable options in practice.

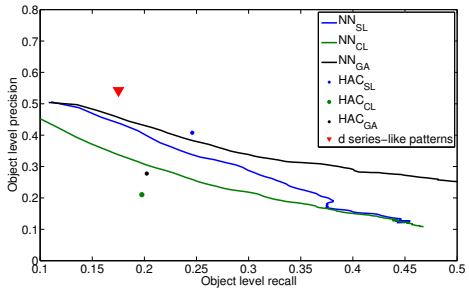

Figure 6: Object-level precision and recall

Case Studies

We performed a blind test, where we aimed to detect crime patterns between 2007 to 2012 for which we do not have pattern data. The results were analyzed by crime analysts. One particularly interesting crime series includes 10 crimes from November 2006 to March 2007. Figure 7 shows geographically where these crimes were located. Table 2 provides some of the details about the crimes within the series.

Considering the (pattern-general) similarity graph, the subgraph containing the 10 crimes is diagrammed in Figure 8. Crimes 1 to 5 are well connected as a subset, and crimes 6 to 10 are well connected as another subset. From only the similarity graph, the two subsets do not seem very related except for a single edge between crimes $\{5, 6\}$ connecting them; however, this is only the pattern-general part of the story.

We used the linear integer programming (2) to discover core sets of size 3 with at least 6 defining features. Table 3 lists the core sets and their defining features in this series, where “1" means the feature is a defining feature and “0" means it is not. These core sets show

Table 2: An example of a series-like pattern with d=6.

NO	Date	Loc of entry	Mns of entry	Premises	Rans	Resid	Time of day	Day	Suspect	Victim
1	11/8/06	Basement Door	Unknown	Unknown	No	Not in	10:45-15:00	Wed	null	1 F
2	11/8/06	Front door	Pried	Unknown	No	Not in	8:00-18:30	Wed	null	1 M
3	11/16/06	Front door	Shoved/Forced	Unknown	No	Not in	9:00-17:00	Thur	null	1 M
4	12/7/06	Front door	Pried	Unknown	No	Not in	9:00-17:00	Thur	null	1 F
5	12/22/06	Front door	Pried	Unknown	No	In	11:48	Fri	null	1 M
6	2/1/07	Front door	Shoved/Forced	Unknown	No	In	14:45	Thur	3 Males	1 F
7	2/15/07	Front door	Unknown	Aptment	No	In	12:00-13:30	Thur	null	2 F
8	3/5/07	Front door	Shoved/Forced	Aptment	No	Not in	12:22-14:56	Mon	null	White M
9	3/5/07	Front door	Broke	Aptment	No	Not in	12:22-14:56	Mon	null	1 F & 1 M
10	3/8/07	Front door	Pried	Aptment	No	Not in	12:50-13:30	Thur	null	1 M

Table 3: Core sets and their defining features for example 1

Core Sets	Crimes	Geo Loc	Days apart	Loc of entry	Mns of entry	Premises	Rans	Resid	Time of day	Day	Suspect	Victim
1	1 2 3	✓	✓	✓	0	*O*	✓	*O*	✓	✓	*O*	*O*
2	1 2 4	✓	✓	✓	*O*	*O*	✓	*O*	✓	✓	*O*	*O*
3	1 2 5	✓	✓	✓	*O*	*O*	✓	*O*	✓	✓	*O*	*O*
4	1 3 4	✓	✓	✓	*O*	*O*	✓	*O*	✓	✓	*O*	*O*
5	1 3 5	✓	✓	✓	*O*	*O*	✓	*O*	✓	✓	*O*	*O*
6	1 4 5	✓	✓	✓	*O*	*O*	✓	*O*	✓	✓	*O*	*O*
7	1 5 6	✓	✓	✓	*O*	*O*	✓	*O*	✓	✓	*O*	*O*
8	2 3 4	✓	✓	✓	*O*	*O*	✓	*O*	✓	✓	*O*	*O*
9	2 3 5	✓	✓	✓	*O*	*O*	✓	*O*	✓	✓	*O*	✓
10	2 4 5	✓	✓	✓	✓	*O*	✓	*O*	✓	✓	*O*	*O*
11	2 4 6	✓	✓	✓	*O*	*O*	✓	*O*	✓	✓	*O*	*O*
12	2 5 6	✓	✓	✓	*O*	*O*	✓	*O*	✓	✓	*O*	*O*
13	3 4 5	✓	✓	✓	*O*	*O*	✓	*O*	✓	✓	*O*	*O*
14	3 5 6	✓	✓	✓	*O*	*O*	✓	*O*	✓	✓	*O*	*O*
15	4 5 6	✓	✓	✓	*O*	*O*	✓	*O*	✓	✓	*O*	*O*
16	5 6 7	*O*	✓	✓	*O*	*O*	✓	✓	✓	✓	*O*	*O*
17	6 8 9	✓	✓	✓	*O*	*O*	✓	*O*	✓	✓	*O*	*O*
18	6 8 10	✓	✓	✓	*O*	*O*	✓	*O*	✓	✓	*O*	*O*
19	6 9 10	✓	✓	✓	*O*	*O*	✓	*O*	✓	✓	*O*	*O*
20	7 8 9	✓	✓	✓	*O*	✓	✓	*O*	✓	✓	*O*	*O*
21	7 8 10	✓	✓	✓	*O*	✓	✓	*O*	✓	✓	*O*	*O*
22	7 9 10	✓	✓	✓	*O*	✓	✓	*O*	✓	✓	*O*	*O*
23	8 9 10	✓	✓	✓	*O*	✓	✓	*O*	✓	✓	*O*	*O*

how the crimes are similar to each other in a pattern-specific way. The next step is merging the core sets. The defining feature set $\Pi(G)$ was chosen to include 6 features, which are geographic location, days apart, location of entry, the ransacked indicator, time of day, and day of the week. One of the core sets, core set 16 in Table 3, was not included as geographic location is not a defining feature for that core set, and the rest of the core sets were merged.

As these data were reconsidered by crime analysts, we found out that when these crimes were analyzed back in 2006-2007, they were viewed as two unrelated patterns, one at the end of 2006, crimes 1 to 5, and one at the beginning of 2007, crimes 6 to 10. The connection between these two subsets of crime is very subtle and there is over a month gap between the two patterns, so it did not occur to the crime analysts to link them. Their intuition agrees completely with the similarity graph, as the two subsets are weakly connected only by one edge; however, recall that this only describes the pattern-general similarity - what one would expect from generic pattern without considering a specific M.O. On examination of the core sets, not only are they are correlated in 6 features, but five of the core sets (core set indices 11, 12, 14, 15, 16) contain crimes from both of the subsets, which is strong evidence that the two subsets should be merged together. Analysts now believe that these two series were actually a single series. It is particularly interesting that the core set consisting of crimes 5, 6, and 7 spanned the two subsets, where these crimes share the unusual feature that residents were present during the break-in. Also one of the crimes has suspect information (3 males) that can be carried through to all the crimes in the discovered series.

This is a good example to show how crime patterns can be composed of core sets, and exhibit similarity both in a pattern-general way and pattern specific way. It shows how we can use both aspects to mine patterns.

Table 4 and Figure 9 shows a more typical pattern in 2007 discovered by our method. According to the Cambridge police, they arrested a suspect and confirmed that he did commit all these crimes. The offender was employed by a realty company who managed some of the buildings where break-ins occurred. The offender was familiar with the buildings and knew the residents' schedule, hence he was able to commit several crimes on the same day, within a short timespan.

Acknowledgements. This work was supported by a

Table 4: Data from second crime series

NO	Date	Loc of entry	Mns of entry	Premises	Rans	Resid	Time of day	Day	Suspect	Victim
1	1/25/07	Front Door	Punched/Popped	Apartment	No	Not in	10:20-12:00	Thur	null	1 F
2	1/25/07	Unknown	Cut Screen	Apartment	No	Not in	8:45-14:30	Thur	null	1 M
3	1/25/07	Unknown	Pried	Apartment	No	Not in	9:10-21:00	Thur	null	1 F
4	1/25/07	Front door	Unlocked	Apartment	No	In	13:00-13:30	Mon	null	1 F
5	1/29/07	Front door	Key	Apartment	No	Not in	14:52-14:52	Mon	null	2 M & 3 F
6	1/29/07	Unknown	Unknown	Apartment	No	Not in	12:00-12:00	Mon	1 M	1 M
7	1/29/07	Unknown	Unknown	Apartment	No	Not in	15:00-15:00	Mon	null	2 M

Figure 7: The locations of crimes in first series.

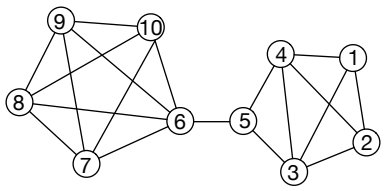

Figure 8: Similarity graph for first crime series

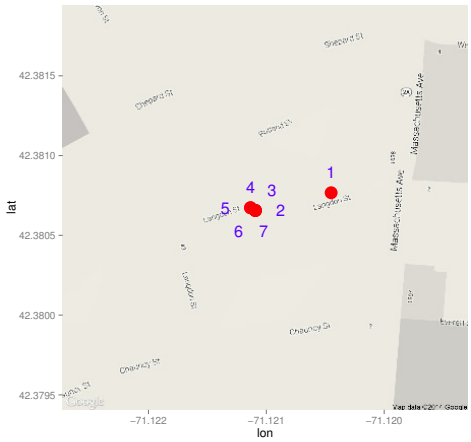

Figure 9: The locations of crimes in second series.

grant from MIT Lincoln Laboratories. Thanks to Hoda Bidkhori for helpful discussions.

References

Brown, D. E., and Hagen, S. 2003. Data association methods with applications to law enforcement. *Decision Support Systems* 34(4):369–378.

Dahbur, K., and Muscarello, T. 2003. Classification system for serial criminal patterns. *Artificial Intelligence and Law* 11(4):251–269.

Domeniconi, C.; Papadopoulos, D.; Gunopulos, D.; and Ma, S. 2004. Subspace clustering of high dimensional data. In *SDM*.

Ghahramani, Z., and Heller, K. 2005. Bayesian sets. In *Proc. NIPS*.

Günnemann, S.; Boden, B.; and Seidl, T. 2011. DB-CSC: a density-based approach for subspace clustering in graphs with feature vectors. In *Proc. ECML-PKDD*, 565–580. Springer.

Gwinn, S. L.; Bruce, C.; Cooper, J. P.; and Hick, S. 2008. Exploring crime analysis. Readings on essential skills, Second edition. Published by BookSurge, LLC.

Hastie, T.; Tibshirani, R.; Friedman, J.; and Franklin, J. 2005. *The elements of statistical learning: data mining, inference and prediction.* Springer.

Kelling, G. L.; Pate, A. M.; Dieckman, D.; and Brown, C. 1974. The kansas city preventive patrol experiment. *Technical report. Washington, DC: Police Foundation.*

Kriegel, H.-P.; Kröger, P.; and Zimek, A. 2009. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM TKDD* 3(1):1.

Letham, B.; Rudin, C.; and Heller, K. A. 2013. Growing a list. *Proc. ECML-PKDD* 27(3):372–395.

Lin, S., and Brown, D. E. 2003. An outlier-based data association method for linking criminal incidents. In *Proc. SDM*.

Moser, F.; Colak, R.; Rafiey, A.; and Ester, M. 2009. Mining cohesive patterns from graphs with feature vectors. In *SDM*, volume 9, 593–604.

Nath, S. V. 2006. Crime pattern detection using data

mining. In *Proc. Web Intelligence and Intelligent Agent Technology Workshops*, 41–44.

Neill, D. B., and Cooper, G. F. 2010. A multivariate bayesian scan statistic for early event detection and characterization. *Machine learning* 79(3):261–282.

Pearsall, B. 2010. Predictive policing: The future of law enforcement? *National Institute of Justice Journal* 266:16–19.

Pei, J.; Zhang, X.; Cho, M.; Wang, H.; and Yu, P. S. 2003. Maple: A fast algorithm for maximal pattern-based clustering. In *ICDM 2003*, 259–266. IEEE.

Sherman, L., and Eck, J. 2002. Policing for crime prevention. *Evidence-Based Crime Prevention*.

Sherman, L. W.; Garlin, P. R.; and Buerger, M. E. 1989. Hot spots predatory crime: routine activities and the criminology of place. *Criminology* 27(1):27–56.

Vidal, R. 2011. Subspace clustering. *Signal Processing Magazine, IEEE* 28(2):52–68.

Wang, H.; Wang, W.; Yang, J.; and Yu, P. S. 2002. Clustering by pattern similarity in large data sets. In *Proc.ACM SIGMOD*, 394–405.

Wang, T.; Rudin, C.; Wagner, D.; and Sevieri, R. 2013. Learning to detect patterns of crime. In *Proc. ECML-PKDD*.

Weisburd, D., and Mazerolle, L. G. 2000. Crime and disorder in drug hot spots: Implications for theory and practice in policing. *Police Quarterly* 3(3):331–349.

Weisburd, D. 2012. Bringing social context back into the equation. *Criminology and Public Policy* 11(2):317–326.

Weisel, D. L. 2002. Burglary of single-family houses. Problem-Oriented Guides for Police Series, No.18.