HARMONIC SPACE-TIME THREAT PROPAGATION FOR GRAPH DETECTION

Steven T. Smith,† Scott Philips,† Edward K. Kao†‡

†MIT Lincoln Laboratory; 244 Wood Street; Lexington MA USA 02420
‡Department of Statistics, Harvard University; Cambridge MA USA 02138
{ stsmith, scott.philips }@ll.mit.edu, edwardkao@fas.harvard.edu

ABSTRACT

This paper addresses threat propagation on space-time graphs, defined to be a time-sampled graph. The application considered is geographical sites connected by tracks, though such graphs arise in many fields. Several new concepts and efficient algorithms are introduced, specifically, the space-time adjacency matrix and harmonic threat propagation. The cued threat propagation problem is shown to be equivalent to the harmonic solution to Laplace's equation on the graph. Alternately, the Perron-Frobenius theorem is applied to a modified space-time adjacency matrix to derive a concept of eigen-threat on space-time graphs. Both approaches yield fast, scalable algorithms for space-time threat propagation applicable to both very small and very large graphs. Algorithms are motivated by a continuous time stochastic process model. Detection performance is shown using a simulated insurgent network data for which harmonic space-time threat propagation achieves an 84% probability of detection with a 4% false alarm probability over the entire graph.

1. INTRODUCTION

Graph exploration and detection are the common objectives in a wide variety of applications ranging from social network analysis, web tracking and advertising, law enforcement, and counter-terrorism. In the case where connections between the graph's vertices are dynamic, temporal correlations may be used to improve the performance of graph detection. This paper considers the problem of "threat propagation" motivated by the example of geographic sites connected together by a set of time-stamped tracks [5, 6, 10]; this particular formulation is analogous to the centrality metrics commonly used in network analysis [4, 7, 8]. Several new concepts and efficient algorithms are introduced that address the problems of exploration and detection in graphs with both spatial and temporal characteristics—space-time graphs. A space-time graph is defined here to be a time-sampled graph with a specific edge set. Also defined are the corresponding concepts of the space-time adjacency matrix and space-time threat propagation. A continuous time stochastic process model is used to motivate the use of algebraic rules for threat propagation. It is shown that in the case of "cued" threat propagation—a priori threat assigned to one or more vertices—the threat propagation problem may be viewed as the harmonic solution to Laplace's equation on the graph, and the corresponding algorithm is called harmonic threat propagation. Alternately, a related "eigen-threat" algorithm is derived based on the stochastic model and the Perron-Frobenius theorem. Many graphs from practical applications are quite large, containing thousands of vertices or more, and their corresponding time-sampled space-time graphs are very large, easily exceeding size one million. The approaches and algorithms developed in this paper all scale from very small space-time graphs, such as those encountered at the beginning of graph exploration, to very large space-time graphs when the full graph is known.

†This work is sponsored by the National Geospatial-Intelligence Agency under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions and recommendations are those of the author and are not necessarily endorsed by the United States Government.

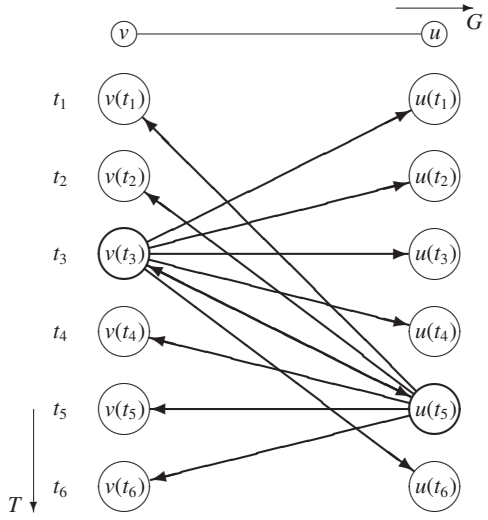

Fig. 1. An example of a directed space-time graph G_T with vertices $V \times T$, i.e., $V = \{u, v\}$ sampled at index times $T = (t_1, \ldots, t_6)$.

2. SPACE-TIME GRAPHS AND PROPAGATION

2.1. Space-Time Graphs and Adjacency Matrices

The set and graph theoretic notation of Diestel [1] and Godsil and Royle [3] will be used, where $V \times T$ denotes the Cartesian product of sets V and T, and $[V]^2 \subset 2^V$ denotes the set of all 2-element subsets of V. Let $G = (V, E)$ be a *simple graph* with vertex set V and edge set $E \subset [V]^2$. The *adjacency matrix* $\mathbf{A} = \mathbf{A}(G)$ of G is the $(0, 1)$-matrix with $\mathbf{A}_{ij} = 1$ iff $\{i, j\} \in E$. A *directed graph* G^σ (also denoted G by abuse of notation) is determined by an orientation $\sigma\colon [V]^2 \to V \times V$ (the ordered Cartesian product of V with itself). A *weighted directed graph* is a directed graph G^σ along with a map $a\colon V \times V \to \mathbb{R}$ that assigns a weight a_{ij} to each arc from node v_i to node v_j. The *weighted adjacency matrix* is simply the matrix $\mathbf{A}_{ij} = a(v_i, v_j)$.

Definition 1 (Space-Time Graph and Adjacency Matrix) *(1) Let $G = (V, E)$ be either a simple or a directed graph, and $T = (t_1, t_2, \ldots, t_K)$ be an index set of K sample times. A* space-time graph

$$G_T = (V \times T, E_T) \tag{1}$$

is defined by the vertices formed by the Cartesian product $V \times T = T \coprod \cdots \coprod T$ (V sampled at times T, or the disjoint union of T with

itself #V times) and an edge set $E_T \subset [V \times T]^2$ *that satisfies the two constraints:*

a) If $(u(t_k), v(t_l)) \in E_T$ *then* $(u, v) \in E$.

b) All temporal subgraphs $((u, v), E_T(u, v))$ *between any two nodes u and v are defined by a temporal model* $E_T(u, v) \subset [T \coprod T]^2$.

A directed space-time graph G_T^σ *is a space-time graph* G_T *along with an orientation* σ *for each of the edges. A* weighted space-time graph *is a directed space-time graph with along with a map* $a\colon (V \times T) \times (V \times T) \to \mathbb{R}$ *that assigns a weight* $a_{i;kl}$ *to each arc from node* $v_i(t_k)$ *to node* $v_j(t_l)$*. (2) The* weighted space-time adjacency matrix *of the space-time graph* G_T *is the weighted adjacency matrix* $\mathbf{A}_{i;kl} = a(v_i(t_k), v_j(t_l))$.

In this definition of a space-time graph, the temporal model $E_T(u, v) \subset [T \coprod T]^2$ is determined by the temporal correlations between vertices at specific times, and is therefore dependent on the specifics of the problem (Fig. 1). A concrete example is provided in Section 2.2.

2.2. Space-Time Threat Propagation

Let $G = (V, E)$ be a graph whose vertices are connected by tracks with known start and stop times. A track τ departs vertex v at time t_τ^v and arrives at vertex u at time t_τ^u; assume that all times are discretized from the index set $T = (t_1, \ldots, t_K)$. Given a known quantity such as "threat" at a particular node, these tracks induce threat propagation through the graph over time. The threat propagation model will determine a space-time graph $G_T = (V \times T, E_T)$.

Though a discrete set of index times T will be used throughout the paper, the rules and functional forms for threat propagation around the graph will be motivated by a continuous time stochastic process model for threat. This model will also be used to establish the use of algebraic rules for threat propagation. Given a vertex v, denote the threat at v and at time $t \in \mathbb{R}$ by the $\{0, 1\}$-valued stochastic process $\Theta_v(t)$, with value zero indicating no threat, and value unity indicating a threat.

Definition 2 *The* probability of threat *at v at t is given by*

$$\vartheta_v(t) \stackrel{\text{def}}{=} P(\Theta_v(t) = 1) = P(\Theta_v(t)). \tag{2}$$

By abuse of notation, the event $\{\Theta_v(t) = 1\}$ will be written as $\Theta_v(t)$ as in the second part of Eq. (2).

Assume that a cued threat at v at time $t = 0$ with probability θ_1 is a finite-state continuous-time Markov jump process between from state 1 to state 0 with Poisson rate λ_v, i.e., a threat present at v has an expected departure time of λ_v^{-1}. Under this assumption, the threat stochastic process $\Theta_v(t)$ satisfies the Itô stochastic differential equation,

$$d\Theta_v = -\Theta_v \, dN_v; \quad \Theta_v(0) = \theta_1, \tag{3}$$

where $N_v(t)$ is a Poisson process with rate λ_v. Eq. (3) defines a finite-state stochastic system that may transition from value unity ("threat") to zero ("non-threat"), but cannot transition from zero to unity—thus representing threat information from either an explicit cue or from tracks connecting the cued threat to other vertices. Conditioning on $\theta_1 \stackrel{\text{def}}{=} \vartheta_v(0)$ and defining $\theta_0 = 1 - \theta_1$, the probability distribution between the threat at time 0 and at time t is,

$$P(\Theta_v(0), \Theta_v(t)) = \begin{smallmatrix} \Theta_v(t)=0 \\ \Theta_v(t)=1 \end{smallmatrix} \begin{bmatrix} \theta_0 & (1-\exp(-\lambda_v t))\theta_1 \\ 0 & \exp(-\lambda_v t)\theta_1 \end{bmatrix}, \tag{4}$$

with $P(\Theta_v(0))$ defined along the columns and $P(\Theta_v(0))$ along the rows.

Given a threat cue on vertex v at time 0, define the *space-time threat kernel* as probability of threat $\vartheta_v(t)$. A threat that satisfies the Markov jump process of Eq. (3) has threat kernel

$$K_v(t) = e^{-\lambda_v |t|}. \tag{5}$$

Other threat kernels are possible, such as a Gaussian kernel [9].

Now consider the combined threat from threat at vertex u arriving or departing at vertex v at time t along a track. By the addition law of probability, the probability of threat $\vartheta_v(t^\pm) = P(\Theta_v(t^\pm))$ at v at time $t^\pm$ immediately after/before the track from v arrives/departs is modeled by the equation

$$P(\Theta_v(t) \cup \Theta_u(t)) = P(\Theta_v(t)) + P(\Theta_u(t)) - P(\Theta_v(t)\Theta_u(t)). \tag{6}$$

If the threat at u and v is independent, then so are the random variables $\Theta_u(t)$ and $\Theta_v(t)$, hence $P(\Theta_v(t)\Theta_u(t)) = P(\Theta_v(t))P(\Theta_u(t))$. If these probabilities are moderately small, then this term of Eq. (6) is small, yielding the approximation

$$\vartheta_v(t^\pm) = P(\Theta_v(t) \cup \Theta_u(t)) \approx P(\Theta_v(t)) + P(\Theta_u(t)) = \vartheta_v(t) + \vartheta_u(t). \tag{7}$$

If the threat at u and v is dependent according to the Poisson process of Eq. (4) with a cued threat at vertex v^* and tracks leading from v^* to both u and v with propagation time difference $\Delta t > 0$, then setting $\theta_1 = P(\Theta_u(t))$, we have $P(\Theta_v(t)) = e^{-\lambda_{v^*}\Delta t}\theta_1$ and $P(\Theta_v(t)\Theta_u(t)) = e^{-\lambda_{v^*}\Delta t}\theta_1$. Therefore, $P(\Theta_v(t) \cup \Theta_u(t)) = \theta_1 + e^{-\lambda_{v^*}\Delta t}\theta_1 - e^{-\lambda_{v^*}\Delta t}\theta_1 = \theta_1$. For large time differences relative to the Poisson time $\lambda_{v^*}^{-1}$, the expression $e^{-\lambda_{v^*}\Delta t}$ is small, and we may again apply the linear approximation of Eq. (7) for the combination of independent threats. For small time differences—especially in the case of vertices adjacent to a cue node—this approximation does not hold and must be treated separately in a linear representation of threat propagation. In the case of many tracks arriving at a vertex at nearby times, the inclusion-exclusion principle and these same arguments provide a linear approximation for threat propagation:

$$\begin{aligned} &P(\Theta_{v_1}(t) \cup \Theta_{v_2}(t) \cup \cdots \cup \Theta_{v_d}(t)) = \textstyle\sum_k P(\Theta_{v_k}(t)) \\ &- \textstyle\sum_{kl} P(\Theta_{v_k}(t)\Theta_{v_l}(t)) + \cdots + (-1)^{d-1} P(\textstyle\prod_k \Theta_{v_k}(t)) \approx \textstyle\sum_k \vartheta_{v_k}(t) \end{aligned} \tag{8}$$

Eqs. (7) and (8) may be conditioned on the probability that threat traverses a particular edge, i.e. $P(\Theta_v(t)) = \sum_k P(\Theta_v(t)|v_k \to v)P(v_k \to v)$, where $P(v_k \to v)$ is the prior. It will be convenient to model this prior as an unnormalized weight function $w\colon G \to \mathbb{R}$ on the graph. Summarizing, the probability of threat $\vartheta_v(t)$ at a vertex v connected by tracks arriving/departing at time t from vertices $v_1, v_2, \ldots, v_d$ is approximated by the linear relationship

$$\vartheta_v(t) \approx (Cw(v))^{-1}(\vartheta_{v_1}(t) + \vartheta_{v_2}(t) + \cdots + \vartheta_{v_d}(t)) \tag{9}$$

where C is the normalization constant. Clearly, the space-time adjacency matrix plays a central role in determining threat propagation. Furthermore, Eq. (9) is recognized has having the form of a mean-value property on the graph, whose solutions may be viewed as discretized harmonic functions, i.e. solutions to Laplace's equation for a certain Laplace-Beltrami operator [11]. This point of view will be essential in the motivation of threat propagation algorithms.

The model of how threat propagates from vertex u to vertex v over time is now presented, at first ignoring any threat that may be present at v. A track departs u at time t_τ^u and arrives at t_τ^v; therefore, by the Poisson process model of Eq. (4), the time-dependent threat at v is given by the equation

$$\vartheta_v(t) = \vartheta_u(t_\tau^u)K(t - t_\tau^v), \tag{10}$$

$$= \textstyle\int_{-\infty}^{\infty} K(t - t_\tau^v)\delta(\sigma - t_\tau^u)\vartheta_u(\sigma)\, d\sigma, \tag{11}$$

where $K_v(t) = e^{-\lambda_v|t|}$ as in Eq. (5). Note that threat propagation is a linear operator from the space of temporal functions at a vertex to another vertex. Discretizing time, the temporal matrix $\mathbf{K}_\tau^{uv}$ for this discretized operator has the sparse form

$$\mathbf{K}_\tau^{uv} = \begin{pmatrix} \mathbf{0} & \ldots & \mathbf{0} & K(t_k - t_\tau^v) & \mathbf{0} & \ldots & \mathbf{0} \end{pmatrix}, \tag{12}$$

where $\mathbf{0}$ represents an all-zero column, t_k represents a vector of discretized time, and the discretized function $K(t_k - t_\tau^v)$ appears in the column corresponding to the discretized time at t_τ^u. Threat propagating from vertex v to u along the same track τ is given by the comparable expression $\vartheta_u(t) = \vartheta_v(t_\tau^v)K(t - t_\tau^u)$ whose discretized linear operator $\mathbf{K}_\tau^{vu}$ takes the form $\mathbf{K}_\tau^{vu} = \begin{pmatrix} \mathbf{0} & \dots & \mathbf{0} & K(t_k - t_\tau^u) & \mathbf{0} & \dots & \mathbf{0} \end{pmatrix}$ [cf. Eq. (12)] where the nonzero column corresponds to t_τ^v. The sparsity of $\mathbf{K}_\tau^{uv}$ and $\mathbf{K}_\tau^{vu}$ will be essential for practical space-time threat propagation algorithms.

Theorem 1 *Let $G = (V, E)$ be a graph sampled at times T, and any two vertices u and v connected by tracks $\tau_1, \dots, \tau_L$ with matrices $\mathbf{K}_{\tau_l}^{uv}$ and $\mathbf{K}_{\tau_l}^{vu}$ as in Eq. (12). The space-time graph $G_T = (V \times T, E_T)$ is defined by the weighted space-time adjacency matrix $\mathbf{A}$, whose submatrix $\mathbf{A}_{uv}$ corresponding to vertices u and v is given by the weighted space-time adjacency matrix*

$$\mathbf{A}_{uv} = \begin{pmatrix} \mathbf{0} & \sum_l \mathbf{K}_{\tau_l}^{vu} \\ \sum_l \mathbf{K}_{\tau_l}^{uv} & \mathbf{0} \end{pmatrix}. \tag{13}$$

3. HARMONIC THREAT ON SPACE-TIME GRAPHS

3.1. Space-Time Laplacian and Harmonic Threat

This section describes memory and compute-efficient algorithms to compute cued space-time threat propagation. These methods are based on standard matrix inversion and that the approximation for threat in Eq. (9) satisfies the mean-value property, i.e. threat is a harmonic function for a Laplace-Beltrami operator.

Definition 3 *(1) Let $\mathbf{A}$ be the space-time adjacency matrix for the space-time graph G_T. Given a weight $w\colon G \to \mathbb{R}$ for each vertex in G with $\mathbf{D} = \mathrm{diag}(w(v_1)\mathbf{I}, \dots, w(v_N)\mathbf{I})$, the* space-time Laplacian $\mathbf{L}$ *is the matrix*

$$\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{A}. \tag{14}$$

(2) Given a cue at vertices $v_{b_1}, \dots, v_{b_C}$, the harmonic space-time threat propagation equation *is*

$$\begin{pmatrix} \mathbf{L}_{\mathrm{ii}} & \mathbf{L}_{\mathrm{ib}} \end{pmatrix} \begin{pmatrix} \vartheta_{\mathrm{i}} \\ \vartheta_{\mathrm{b}} \end{pmatrix} = \mathbf{0} \tag{15}$$

where the space-time Laplacian $\mathbf{L} = \begin{pmatrix} \mathbf{L}_{\mathrm{ii}} & \mathbf{L}_{\mathrm{ib}} \\ \mathbf{L}_{\mathrm{bi}} & \mathbf{L}_{\mathrm{bb}} \end{pmatrix}$ and the space-time threat vector $\boldsymbol{\vartheta} = \begin{pmatrix} \vartheta_{\mathrm{i}} \\ \vartheta_{\mathrm{b}} \end{pmatrix}$ have been permuted so that cued vertices are in the 'b' blocks (the "boundary"), non-cued vertices are in 'i' blocks (the "interior"), and the cued space-time vector ϑ_{b} is given. (3) The harmonic threat *is the solution to Eq. (15),*

$$\vartheta_{\mathrm{i}} = -\mathbf{L}_{\mathrm{ii}}^{-1}(\mathbf{L}_{\mathrm{ib}}\vartheta_{\mathrm{b}}). \tag{16}$$

Note that the space-time Laplacian is a so-called directed Laplacian matrix, and that Eq. (15) [cf. Eq. (9)] is directly analogous to Laplace's equation $\Delta\varphi = 0$ given a fixed boundary condition. [11] The weights in $\mathbf{D}$ may be the degree of each vertex, i.e. $w(v) = d(v)$, or the generalized Dijkstra distance from a cue vertex v^*, i.e. $w(v) = \mathrm{dist}(v, v^*)$ where dist is the smallest sum of degrees along a path from v^* to v, thereby mitigating the effect of paths through high-degree vertices. The generalized Dijkstra distance will be used for threat propagation in the results of the next section. The space-time adjacency matrix $\mathbf{A}$ and cued threat vector ϑ_{b} are nonnegative; therefore, the harmonic threat of Eq. (16) is also nonnegative. The biconjugate gradient method can be used to solve this highly sparse linear system, providing a practical computational approach for space-time threat propagation. This approach scales well to graphs with thousands of vertices and thousands of time samples, resulting in space-time graphs of order ten million or more. In practice, significantly smaller subgraphs are encountered in applications such as threat network discovery [10], for which linear solvers with sparse systems are extremely fast.

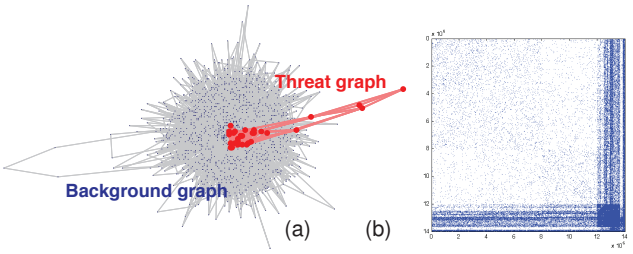

Fig. 2. (a) Graph of the NGA simulated data comprised of 4 478 vertices and 116 720 tracks. The foreground subgraph is shown using red vertices and edges, and the background graph is shown using blue vertices and gray edges. For clarity, the full graph has been downsampled by a factor of four. (b) Space-Time adjacency matrix of the NGA simulated data with 313 time samples. Each dot in this sparse matrix itself represents a sparse 313-by-313 temporal adjacency matrix.

3.2. Eigen-Threat on Space-Time Graphs

Alternatively to harmonic threat, eigenvector centrality [7] may also be used to compute space-time threat propagation [9], which also relies on the form for threat in Eq. (9) as well as the Perron-Frobenius theorem [2, 3]. Assuming an initial estimate $\boldsymbol{\vartheta}^{(0)}$ for the probability of threat across nodes, by Eq. (9) improved estimates are obtained via the sequence $\boldsymbol{\vartheta}^{(k)} = \mathbf{A}\boldsymbol{\vartheta}^{(k-1)} = \mathbf{A}^k\boldsymbol{\vartheta}^{(0)}$ for $k = 1, 2, \dots$. The normalized steady-state solution $\boldsymbol{\vartheta} = \boldsymbol{\vartheta}^{(\infty)}$ is an eigenvector of $\mathbf{A}$. This eigenvector of $\mathbf{A}$ does not account for a cued vertex v^*, for which the threat probability $\vartheta_{v^*}(t)$ is determined by Eq. (4), and as shown by Eq. (6), the approximation used in Eq. (9) does not apply to cued vertices because threat at adjacent vertices is not independent of the cue. Cued vertices may be incorporated by introducing the (weighted) cued space-time adjacency matrix $(\mathbf{D}^{-1}\mathbf{A})^{[v^*]}$, which is the matrix obtained by replacing the block-row corresponding to v^* of $\mathbf{D}^{-1}\mathbf{A}$ with $(\mathbf{0} \dots \mathbf{0}\,\mathbf{I}\,\mathbf{0} \dots \mathbf{0})$, where $\mathbf{D}$ represents a weight on the vertices as in Definition 3. This method is motivated by the block matrix identities $\begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{K} & \mathbf{I} \end{pmatrix}^k = \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ k\mathbf{K} & \mathbf{I} \end{pmatrix}$, $\begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \frac{1}{2}\mathbf{K} & \frac{1}{2}\mathbf{I} \end{pmatrix}^k = \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ (1-2^{-k})\mathbf{K} & 2^{-k}\mathbf{I} \end{pmatrix} \to \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{K} & \mathbf{0} \end{pmatrix}^k = \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{K} & \mathbf{0} \end{pmatrix}$.

The Perron-Frobenius theorem guarantees that if the directed graph G defined by the tracks is strongly connected, then the principal eigenvector of the cued, weighted space-time adjacency matrix $(\mathbf{D}^{-1}\mathbf{A})^{[v_{b_1} \dots v_{b_C}]}$ is also nonnegative, yielding another practical approach to space-time threat propagation:

$$\boldsymbol{\vartheta} = \lambda(\mathbf{D}^{-1}\mathbf{A})^{[v_{b_1} \dots v_{b_C}]}\boldsymbol{\vartheta}. \tag{17}$$

An Arnoldi iteration method such as that used in Matlab's `eigs` command can be used to compute the principal eigenvector. Arnoldi iteration is more slightly expensive (though of comparable complexity) than linear solvers such as the biconjugate gradient method. Therefore, harmonic threat propagation of Section 3.1 is preferred. Furthermore, there is not a significant performance difference between harmonic and eigen-threat observed in the example data used in the next section.

4. DETECTION ON SPACE-TIME GRAPHS

The space-time threat propagation algorithms developed in the previous sections can be used for both prioritized discovery of threat networks [10] and for threat network detection [9, 10]. In this section, detection performance results the later application will be shown simulated vehicle motion data from the National Geospatial-Intelligence Agency (NGA). This data is derived from a scripted scenario that contains a clandestine insurgent network [Fig. 2(a)]. 31 (of 4 478) locations belong to the threat network. Time is discretized into 10 minute intervals, resulting in a space-time adjacency matrix whose order is

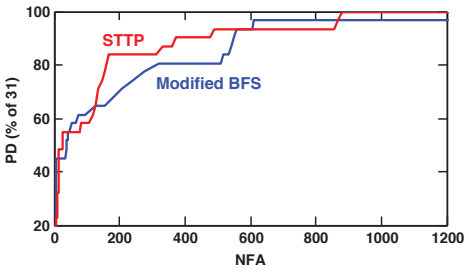

Fig. 3. The probability of detection (PD) versus the number of false alarms (NFA) for the space-time threat propagation (STTP) algorithm (red) and the modified breadth-first search (BFS) algorithm (blue) given the cued vertex shown in Fig. 4.

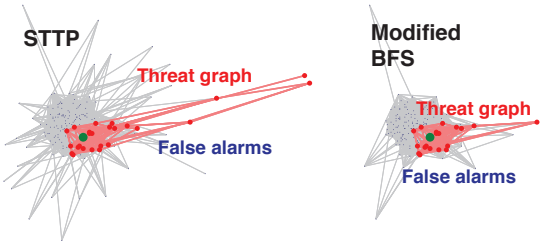

Fig. 4. The detected threat graph for the STTP and modified BFS algorithms at a constant false alarm rate (CFAR) at NFA = 180. The cue is green, the truth is red, and the false alarms are blue. The higher detection rate of PD = 84% for STTP compared to 65% for BFS is observed, resulting in a greater number of threat vertices detected.

over 1.4 million [Fig. 2(b)]. Forming this sparse matrix requires about 1 GiB and 15 seconds on an x86_64 GHz dual-core laptop computer. Computing the harmonic threat takes 5 seconds, and the eigen-threat takes 20 seconds. Arnoldi iteration is slightly more expensive that the biconjugate gradient method. Because the compute and memory cost of both methods is $O(N{\cdot}\#E)$ with $N = \#V{\cdot}\#T$ for space-time graphs, all these values scale linearly with the spatial and temporal scale.

Graph detection performance over the entire space-time data set provides a baseline measure of algorithm performance. The receiver operating characteristic (ROC) for the harmonic space-time threat propagation (STTP) algorithm relative to the Dijkstra-weighted adjacency matrix is shown in Fig. 3 for the entire graph, given the cue shown in Fig. 4. For comparison against a nominal spatial-only algorithm, the threat computed using a modified breadth-first search (BFS) algorithm is also shown. Standard BFS uses the principal eigenvector of the degree-weighted adjacency matrix, which is trivially the uniform vector $\vartheta_{\text{BFS}} = (1, 1, \ldots, 1)^{\text{T}}$, i.e. threat diffuses evenly to vertices connected via tracks to the cued node. This trivial solution is avoided by taking threat computed from the harmonic threat of the Dijkstra-weighted spatial adjacency matrix. The amount by which the performance of the STTP algorithm exceeds the spatial-only BFS algorithm shows the detection improvement gained by using temporal information.

There are two steps to cued network detection: network discovery and network detection [10]. Network discovery involves exploring and building a graph over time based on cued information and search prioritization. Network detection involves a binary decision of discovered vertices into threats and nonthreats. Space-Time threat propagation improves the performance of the network discovery step over spatial-only algorithms because temporal information affects the prioritization of graph exploration. If the entire graph has been explored, temporal information may also be useful in the detection step that involves the computation of threat at each vertex. The relative benefit of temporal information for graph detection is expected to be greater for smaller graphs and smaller time extents, though this will not be quantified in the paper. The space-time algorithm outperforms the spatial-only algorithm over most of the ROC curve in Fig. 3. At NFA = 180, the STTP algorithm has a detection probability of 84% and the modified BFS algorithm has 65%. At the fixed probability of detection PD = 84% the STTP algorithm has 180 false alarms and BFS has 517. The performance of the eigen-threat algorithm of Section 3.2 is also computed, but it is statistically (and graphically) indistinguishable from the performance of the harmonic threat algorithm. For graphs much smaller than the one represented by the entire data set, as encountered during network exploration, the detection performance of both algorithms is expected to decrease with an increased relative performance of space-time algorithms over spatial-only algorithms.

5. CONCLUSIONS

Temporal correlations improve the performance of graph exploration and detection applications in which vertices have time-dependent connections. Definitions are introduced for a space-time graph and corresponding notions of the space-time adjacency matrix and space-time threat propagation. For cued threat propagation, the threat propagation problem is equivalent to the harmonic solution to Laplace's equation on the graph; this is called the harmonic space-time threat. Alternately, the principal eigenvector of a modified space-time adjacency matrix also represents space-time threat; this is called eigen-threat. For large graphs, harmonic threat is computed using the biconjugate gradient method, and eigen-threat is computed using Arnoldi iteration, which both scale well over problem size. However, the first method is faster, and the detection performance eigen-threat is statistically indistinguishable from harmonic threat with the data used in Section 4, hence harmonic threat is preferred if only for computational reasons. Both threat propagation algorithms are derived assuming a Poisson continuous time stochastic process and dependency model. The graph detection performance of harmonic space-time threat propagation is shown using simulated vehicle motion data from the National Geospatial-Intelligence Agency and compared to a spatial-only modified breadth-first search method. For a cued threat, the space-time algorithm is shown to exceed the spatial-only method, consistent with comparable results for graph exploration.

6. REFERENCES

[1] R. Diestel. *Graph Theory*. New York: Springer-Verlag, Inc. 2000.

[2] F. R. Gantmacher. *Matrix Theory*. Vol. 2. New York: Chelsea, 1959.

[3] C. Godsil and G. Royle. *Algebraic Graph Theory*. New York: Springer-Verlag, Inc. 2001.

[4] M. O. Jackson. *Social and Economic Networks*, Princeton U. Press, 2008.

[5] B. A. Miller, M. S. Beard, and N. T. Bliss. "Eigenspace analysis for threat detection in social networks," in *Proc. 14th Intl. Conf. Informat. Fusion (FUSION)*, Chicago, IL, July 2011.

[6] B. A. Miller, N. T. Bliss, and P. J. Wolfe. "Subgraph detection using eigenvector L_1 norms," in *Proc. 2010 Neural Information Processing Systems (NIPS)*, Vancouver, Canada, 2010.

[7] M. E. J. Newman. *Networks: An Introduction*, Oxford U. Press, 2010.

[8] M. E. J. Newman. "Finding community structure in networks using the eigenvectors of matrices," *Phys. Rev. E*, **74** (3), 2006.

[9] S. Philips, E. K. Kao, M. Yee, and C. Anderson. "Detecting activity-based communities using dynamic membership propagation," submitted to *IEEE Intl. Conf. Acoustics, Speech and Signal Processing*, 2011.

[10] S. T. Smith, A. Silberfarb, S. Philips, E. K. Kao, and C. Anderson. "Network Discovery Using Wide-Area Surveillance Data," in *Proc. 14th Intl. Conf. Informat. Fusion (FUSION)*, Chicago, IL, July 2011.

[11] T. J. Willmore. "Mean value theorems in harmonic Riemannian spaces," *J. London Math. Soc.* **25** : 54–57, 1950.