# MODELING AND DETECTION OF EVOLVING THREATS USING RANDOM FINITE SET STATISTICS*

Zachariah Sutton Peter Willett Yaakov Bar-Shalom

*Department of ECE*
*University of Connecticut*
*Storrs, Connecticut 06269*

**ABSTRACT**

Many threats in the form of human actions (terrorist attacks, military actions, etc.) can be modeled by someone with relevant expert knowledge. A model would be a hypothesis or guess as to how a threat would develop and what kind of observable evidence it would produce along the way. We present a method of stochastically modeling these types of processes using Hidden Markov Models (HMMs). We then present a detection scheme using a Bernoulli Filter – an increasingly popular application of random finite set statistics

## 1. BACKGROUND

We aim to build a stochastic model of some "interesting" or suspicious pattern of human activity. It is reasonable to assume that a significant threat would involve preliminary activities in the form of transactions (communication, travel, financial transactions, etc.) between entities (people, places, objects) that are involved in the threat. Therefore, the threat can be modeled as a sequence of transactions between an *a priori* unknown set of involved entities, starting with planning steps and culminating at some ultimate goal (attack, successful maneuver, etc.). We assume that some of the preliminary transactions would be observed by thorough intelligence surveillance, along with large amounts of clutter – transactions between entities that play no part in the threat. The challenge we address here is, given such a threat model and a constant stream of intelligence data that is mostly clutter, we want to detect if and when the threat process is present, get an idea of which entities are involved, and ideally tell what stage the threat process is in.

A natural choice for modeling such a process is a Hidden Markov model (HMM). A HMM is a Markov process – discrete in this case – whose states are not directly observable, but have some probabilistic relationship to what is observed. In this case the states would be stages in the threat process, which can't be seen but are expected to be related to the observations. General tutorials on HMMs can be found at [5], [6]. Some previous work has been done on modeling these particular types of threats with HMMs in [8].

Since a modeled threat may or may not be present, a Bernoulli filter lends itself well to the detection problem. The Bernoulli filter uses a random finite set (RFS) framework which maintains a target state estimate that is weighted by the target's probability of existence. A method of tracking threat models using a Bernoulli Filter can be found in [3].

There are two main contributions in this paper. First, we have revised the structure of the HMM transition matrices from previous work. In [8],[3] the observations from the threat process are emitted when a transition occurs from one state to the next, and clutter was emitted any time a state self-transitioned. This reasonably models the real life scenario where we receive relevant observations separated by long periods of clutter. Here we have introduced a different model that achieves much the same effect. In our Markov model structures there are states that can not self-transition and emit the relevant "true" observations. These states are separated by states that have a high chance of self-transitioning and emit clutter observations.

Second, instead of modeling each observation as a single symbol, we consider an observation to be a single transaction symbol connecting two entity symbols. In order to use this model, we have assumed some total population of entities in the observation space that are each uniquely identifiable. This allows us to approach the detection problem as a problem of finding some subset of the total entity population that is performing an anomalous sequence of actions, which is a more specific target than just a sequence of actions. Each entity is given a weight or probability of involvement (0 is definitely not involved, 1 is definitely involved). It is initially assumed that we have no prior knowledge about the entities' weights. We present a method of updating these weights so that the filter acquires more knowledge about the entities as the process progresses which in turn helps with detection. Qualitatively speaking, we use past observations to adjust how "interesting" an entity is, then current observations that involve "interesting" entities are taken more seriously.

*Supported by NPS via ONR contract N00244-16-1-0017.

## 2. MODELS

### 2.1 Population Model

We define an array of all entity identities $\boldsymbol{\mathcal{E}} = \{e_1, e_2, ..., e_{N_e}\}$ of size $N_e$. It is implied that the entities are human actors, but they could just as easily be locations or objects. For each observation, a pair of these entities will be linked by a transaction. This assumes that all entities are uniquely identifiable and will be correctly identified when observed. This requirement could be relaxed in future work if a feature based observation model [8] is used.

With all entities identified, we assign to each an indicator Bernoulli random variable where a value of 1 means the entity is involved in the threat and 0 means it is not involved. Let us denote the distributions (probability of value 1) of these indicator random variables as $\boldsymbol{K} = [k_1, k_2, ..., k_{N_e}]$.

### 2.2 Observation Model

We define a set $\boldsymbol{\mathcal{Z}} = \{z_1, z_2, ..., z_{N_z}, z_\emptyset\}$ of all possible transaction types including a null observation type $z_\emptyset$ that signifies an unintelligible or blank observation. This should be an exhaustive set of the transaction types from clutter and threat processes. We model the current observation $\mathbf{O}_t$ with the structure shown in Figure 1, where $z^t \in \boldsymbol{\mathcal{Z}}$ is some transaction linking entities $\{e_a^t, e_b^t\} \subset \boldsymbol{\mathcal{E}}$.

It is helpful to think of our observation model as an attempt to give structure to simple sentences where $\boldsymbol{\mathcal{Z}}$ is the set of all possible verbs and $\boldsymbol{\mathcal{E}}$ is the set of all possible nouns. In this work we assume that necessary preprocessing of data can be done to fit observations roughly into the framework of this model. This is relatively simple in some cases; e.g. Person A places a call to Person B, or Person X gets on a flight to City J. However, fitting data to this model can be harder in some cases; e.g. Person M posts suspicious social media content, or there is a large crowd of people at Location D.

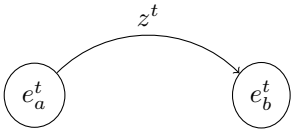


Figure 1: Structure of an observation

### 2.3 Clutter Process Model

A clutter process $\lambda_{cl}$ is modeled as a single state HMM that emits transaction types based on some modeled clutter distribution $p(z|\lambda_{cl})$. This distribution should give some weight to every transaction type in $\boldsymbol{\mathcal{Z}}$ including the null observation. The clutter process also samples two entities uniformly with replacement from $\boldsymbol{\mathcal{E}}$ to be linked by the transaction.

### 2.4 Threat Process Model

One practical modeling detail we must take into consideration is that even if the modeled threat exists, the observations due to our target ("true" observations) will be very sparse – separated by long periods of time where we receive only clutter. This is modeled in [8] and [3] by requiring the target process to output true observations only when there is a transition from one state to the next, and output clutter when states self transition. While this approach is reasonable, it is unnecessarily complicated to implement. For this work we have modified the model of the state transition structure of the HMM to be as shown in Figure 2. Let us denote the set of all light colored states as $\boldsymbol{\mathcal{C}}$, and the set of all shaded states as $\boldsymbol{\mathcal{T}}$. All

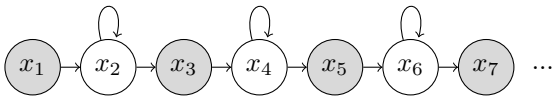


Figure 2: Example Markov chain

states in $\boldsymbol{\mathcal{C}}$ are considered clutter states and are given a very high probability of self transitioning. These states model the long wait periods where the target is not emitting any observations. The states in $\boldsymbol{\mathcal{T}}$ are considered the target states and do not self transition so are only visited once. These states provide the sparse "true" observations that are produced by the target.

This structure allows us to treat the threat model as a typical HMM, eliminating the need to code a transition based model as in [8], [3]. The trade-off is a larger HMM state transition matrix since there are essentially twice as many states as there are in the transition based model. Note that the single path "daisy chain" transition structure is used in this writing for the sake of simplicity, and [3] shows that this structure is indeed the most detectable. But it is not a strict requirement in this work. The transition structure is allowed to split into parallel paths. However, we do require that each of the target states are separated by at least one clutter state.

The state transition matrix $A$ defining the structure in Fig. 2 would have the general form

$$A = \begin{bmatrix} 0 & 1 & 0 & ... & & \\ 0 & p_1 & (1-p_1) & 0 & ... & \\ \vdots & 0 & 0 & 1 & 0 & ... \\ & \vdots & 0 & p_2 & (1-p_2) & 0 & ... \\ & & \vdots & 0 & \ddots & \ddots & \end{bmatrix}$$

where each row is a transition probability distribution for the corresponding state. In other words, the element $A_{i,j}$ is the probability of transitioning from state $i$ to state $j$. Notice that the rows corresponding to target states have 0 on the diagonal and 1 on the superdiagonal, meaning they do not self-

transition and transition immediately to the next state. The rows corresponding to clutter states have some self-transition probability $p_i$ which is not necessarily the same for all clutter states but is set to be very high ($\approx 0.999$).

For the simplest case of one possible transaction type for each target state, the emission matrix $B$ that models the relationship between the states and observed transaction types in $\boldsymbol{\mathcal{Z}}$ will have the general form

$$B = \begin{bmatrix} 0 & 1-p_\emptyset & 0 & ... & 0 & p_\emptyset \\ p_{cl}(z_1) & p_{cl}(z_2) & p_{cl}(z_3) & ... & p_{cl}(z_M) & p_\emptyset \\ 0 & 0 & 0 & ... & 1-p_\emptyset & p_\emptyset \\ p_{cl}(z_1) & p_{cl}(z_2) & p_{cl}(z_3) & ... & p_{cl}(z_M) & p_\emptyset \\ 1-p_\emptyset & 0 & 0 & ... & 0 & p_\emptyset \\ \vdots & \vdots & \vdots & & \vdots & \vdots \end{bmatrix}$$

where each row is the emission distribution over $\boldsymbol{\mathcal{Z}}$ for the corresponding state and must sum to 1.

The observations emitted by target states draw entities independently but exclusively from some unknown population subset $\boldsymbol{I} \subset \boldsymbol{\mathcal{E}}$. Each entity in $\boldsymbol{I}$ is defined as being involved in the threat we are modeling. The algorithm requires an estimate of how many entities will be involved in a given threat $N_{inv}$, but can assume no prior information on the identities.

## 3. BERNOULLI FILTER

We now present a method of detecting and tracking the threat with a Bernoulli filter while simultaneously updating the involvement probabilities in $\boldsymbol{K}$. For now, we restrict our discussion to the case where there is either one target or no target. Ongoing research is exploring ways to detect multiple targets. A general tutorial on Bernoulli filters can be found in [7]. An BF implementation for our particular purpose is presented in [3] using different modeling techniques.

The Bernoulli filter treats the target as a random finite set $\mathbf{S}$ which is either empty $\emptyset$, or it has cardinality 1 and takes on the value of the current HMM state. We can think of the RFS $\mathbf{S}$ as taking on values in the augmented state space $\mathbf{S} \in \{x_1, x_2, ..., x_{N_x}, \emptyset\}$. The BF outputs an estimated pmf $f(\mathbf{S})$ over this state space. The output can also be thought of as a pmf over the states of the target $\{x_1, x_2, ..., x_{N_x}\}$ that is weighted with a probability of target existence $q$, where $(1-q)$ is the probability that the RFS is empty.

We must choose as parameters the probability of target birth $p_b$ and the probability of target survival $p_s$ from one time step to the next. A target birth distribution $f^b(x)$ must also be chosen, which in this case should only give weight to the first state in the transition structure (we assume the process always starts at the beginning).

### 3.1 Dynamics

The dynamics of the RFS are modeled as a Markov process completely parameterized by the probability of target birth $p_b$, the probability of target survival $p_s$, the birth distribution $f^b(x)$, and the HMM state transition matrix $A$. The transition matrix for $\mathbf{S}$ is

$$\boldsymbol{\Pi} = \left[\begin{array}{c|c} p_s \cdot A & \vec{1} \cdot (1-p_s) \\ \hline p_b \cdot f^b(x) & (1-p_b) \end{array}\right] \tag{1}$$

where $\vec{1}$ is a column vector of ones.

### 3.2 Update Step

Say at time $t$ we are given the predicted pmf based on all previous observations $f_{t|t-1}(\mathbf{S}|\mathbf{O}^{t-1})$ and the current observation $\mathbf{O}_t$. We want to update the pmf based on the current observation. The update is given by

$$f_{t|t}(\mathbf{S}|\mathbf{O}^t) = \frac{\varphi(\mathbf{O}_t|\mathbf{S}) f_{t|t-1}(\mathbf{S}|\mathbf{O}^{t-1})}{\sum_{\mathbf{S}} \varphi(\mathbf{O}_t|\mathbf{S}) f_{t|t-1}(\mathbf{S}|\mathbf{O}^{t-1})} \tag{2}$$

where the state dependent observation likelihood is

$$\varphi(\mathbf{O}_t|\mathbf{S}) = \begin{cases} p(z^t|\lambda_{cl})(\frac{1}{N_e})^2, & \text{if } \mathbf{S} \in \{\boldsymbol{\mathcal{C}}, \emptyset\} \\ p(z^t|\mathbf{S})k(e_a)k(e_b)(\frac{1}{N_{inv}})^2, & \text{if } \mathbf{S} \in \{\boldsymbol{\mathcal{T}}\} \end{cases} \tag{3}$$

The term $p(z^t|\mathbf{S})$ is the state dependent probability of the current transaction type which is readily available from the emission matrix of the target HMM. And $k(e_a)$ and $k(e_b)$ are the involvement probabilities of the two observed entities.

### 3.3 Prediction Step

The predicted pmf for time $t+1$ is based solely on the modeled dynamics of the RFS $\mathbf{S}$. In a programmed implementation where the updated pmf is a column vector $\vec{f}_{t|t}(\mathbf{S}|\mathbf{O}^t)$, the predicted pmf is also a column vector given by multiplication with the transpose of the transition matrix in (1).

$$\vec{f}_{t+1|t}(\mathbf{S}|\mathbf{O}^t) = \boldsymbol{\Pi}^{\mathrm{T}} \vec{f}_{t|t}(\mathbf{S}|\mathbf{O}^t) \tag{4}$$

This prediction is then used in (2) at the next time step.

## 4. INVOLVEMENT PROBABILITIES

The vector $\boldsymbol{K}$ contains the probability of involvement (weight) for each entity. Since these weights are the distributions of a set of Bernoulli random variables, it is reasonable for $\boldsymbol{K}$ to sum to a value close to the expected number of involved entities $N_{inv}$. We model the underlying entity process as an array of independent indicators over the entire population where the expected number of "on" indicators at any time is $N_{inv}$, and the expected amount of time an indicator stays on is the expected target lifetime. If we take

$$C = \frac{N_{inv}}{N_e} \tag{5}$$

to be the fraction of population expected to be involved, and $D$ to be the expected target lifetime, the desired behavior is modeled by a two state Markov chain with transition matrix

$$\boldsymbol{H} = \begin{bmatrix} 1 - \frac{1}{D} & \frac{C}{D(1-C)} \\ \frac{1}{D} & 1 - \frac{C}{D(1-C)} \end{bmatrix} \tag{6}$$

where the first state is the involved or "on" state and the second is the uninvolved or "off" state. In simulations, this process does not govern the true entity involvements; instead an involved set is chosen at the time of target birth and remains the same throughout. However, the stationary distribution for the "on" state is $C$ which is also used as the initial weight for all entities. Then the two cases in (3) differ only by the same factor that the terms $p(z^t|\lambda_{cl})$ and $p(z^t|\mathbf{S})$ differ by.

We have devised a method to update the entity weights in $\boldsymbol{K}$ based on the current estimate given by the Bernoulli filter. For the current time step, the pmf over $\mathbf{S}$ is first updated using (2) and the prior entity weights. Then the weights of the two currently observed entities are updated using the formula

$$k'(e_i) = \frac{k(e_i)\left(\frac{1}{N_{inv}} \sum_{s \in \mathcal{T}} f(s) + \frac{1}{N_e} \sum_{s \in \mathcal{C}, \emptyset} f(s)\right)}{\left(k(e_i)\frac{1}{N_{inv}} \sum_{s \in \mathcal{T}} f(s)\right) + \left(\frac{1}{N_e} \sum_{s \in \mathcal{C}, \emptyset} f(s)\right)}, i = a, b \tag{7}$$

where $f(s)$ is shorthand for the value of the updated pmf corresponding to state $s$ and $k(e_i)$ is the prior weight of entity $i$. The entities' updated weights are proportional to how "interesting" their current transaction is and how often they have been linked to "interesting" transactions in the past.

Then $\boldsymbol{H}$ is applied independently to every entity weight including the two current ones updated by (7). Since we are only interested in the weights and not their complements, this update amounts to

$$\hat{k}(e_i) = \left(1 - \frac{1}{D}\right) k(e_i) + \frac{C}{D(1-C)}(1 - k(e_i)), \quad \forall i \tag{8}$$

## 5. EXPERIMENTAL RESULTS

Here we adopt Quickest Detection which is usually used to evaluate the performance of edge detectors or communication system filters [4]. The "signal" we are working with is the probability of target existence output by the BF. Quickest detection quantizes the incoming signal – in this case the probability range 0 to 1 is broken into 100 possible levels. Then each level is treated as if it is the detection threshold. For each threshold level, we mark on the vertical axis the average time interval $\bar{T}$ between points where the threshold is exceeded *while the filter is run on only clutter*. The horizontal value of the data point is the average delay to detection $\bar{D}$ *when the target is present*. This is the average time delay between true target birth time and the time that the existence probability exceeds the threshold. These results are obtained over 2000 Monte Carlo runs (2000 evolutions of the target).

Experimental results are given for a 37 state daisy chain target HMM with an average lifetime of 1550 time steps. This implies that we receive at most 19 observations from our target and roughly 1500 clutter observations over the target's lifetime. Quickest detection results are given in Figure 3. The probability of target existence for a single instance of the target is plotted in Figure 4. The use of entity weights gives a significant improvement in both cases.

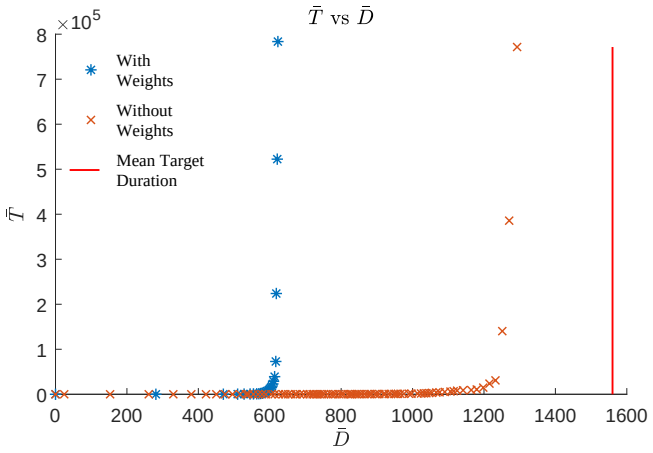


Figure 3: Quickest detection data for the experimental model. Blue data shows results when entity weights are used. Orange results are without. The red line marks the expected target lifetime, past which $\bar{D}$ is somewhat meaningless.

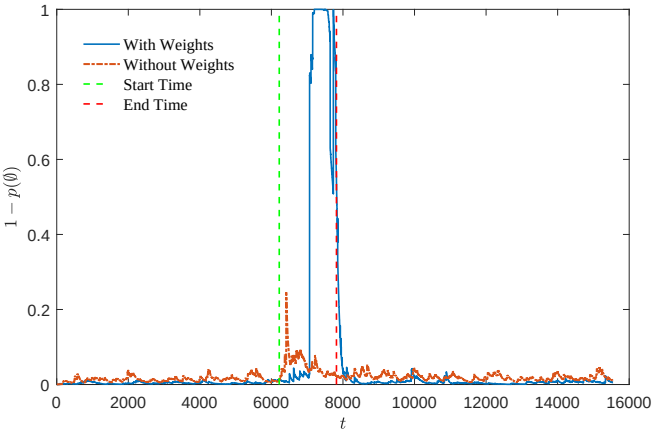


Figure 4: Probability of target existence versus time for one instance of the target. Target start and end times are marked with green and red respectively. Solid blue data shows results when entity weights are used. Dashed orange results are without.

## References


[1] Y. Bar-Shalom, X. R. Li and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software*, J. Wiley and Sons, 2001.

[2] D. Bertsekas, The auction algorithm: A distributed relaxation method for the assignment problem, Annals of Operations Research, vol. 14, no. 1, pp. 105123, 1988.

[3] K. Granstrom, P. Willett, and Y. Bar-Shalom, "Asymmetric Threat Modeling Using HMMs: Bernoulli Filtering and Detectability Analysis," *IEEE Transactions on Signal Processing*, vol. 64, no. 10, pp. 2587-2601, May 2016.

[4] Y. Liu, S. D. Blostein, "Quickest detection of an abrupt change in a random sequence with finite change-time", IEEE Transactions on Information Theory, pp. 1985-1993, November 1994.

[5] L. R. Rabiner and B. H. Juang, "An introduction to hidden Markov models," *IEEE ASSP Mag.*, vol. 3, no. 1, pp. 416, Jan. 1986.

[6] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257286, Feb. 1989.

[7] B. Ristic, B.-T. Vo, B.-N. Vo, and A. Farina, "A Tutorial on Bernoulli Filters: Theory, Implementation and Applications," *IEEE Transactions on Signal Processing*, vol. 61, no. 13, pp. 34063430, Jul. 2013.

[8] S. Singh, H. Tu, W. Donat, K. Pattipati, and P. Willett, "Anomaly detection via feature-aided tracking and hidden Markov models," *Transactions on Systems, Man, and CyberneticsPart A: Systems and Humans*, vol. 39, no. 1, pp. 144159, Jan. 2009.