

Weak process models for robust process detection

Guofei Jiang

Institute for Security Technology Studies and
Thayer School of Engineering, Dartmouth College, Hanover, NH 03755
gfj@dartmouth.edu

ABSTRACT

Many defense and security applications involve the detection of a dynamic process. A process model describes the state transitions of an object, which evolves in time according to specific known laws. Given a process model, the process detection problem is to identify the existence of such a process in large amount of observation data. While Hidden Markov Models (HMMs) are widely used to characterize dynamic processes, it's usually hard to estimate those state transition and emission probabilities precisely in practice, especially if we don't have sufficient training data. An inaccurate process model could lead to high false alarm and misdetection rates and the inference result could be misleading in the decision-making process. To this end, we propose nonparametric weak models derived from HMMs to characterize dynamic processes. A weak model doesn't need the strong requirement for probability specification as in HMMs. In this paper, we analyze the properties of such weak models and propose recursive algorithms to compute the hypotheses of the hidden state sequence and the size of the hypothesis set. Further we analyze how to control the size of the hypothesis set by increasing the number of sensors to tune the structure of the emission matrix.

Keywords: Process detection, inference algorithm, hidden Markov model, nonparametric model, hypothesis

1. INTRODUCTION

Many defense and security applications involve the detection of a dynamic process in large amounts of observation data. For example, the detection of a moving target in battlefield sensor data; the detection of terrorism activity in homeland security intelligence data; the detection of a multi-stage attack in audited data and logs of computer security. We believe that many of these dynamic activities can be described as a deterministic or stochastic dynamic process. A process describes the state transitions of an object, which evolves with time according to specific known laws. For example, the kinematics of a target could be formalized as a state transition equation; the terrorism activity could be described with a Markov model; the multi-stage penetration could be represented in a finite state machine. The process detection problem is to identify the existence of such a process in large amount of observation data with regard to a given process model, i.e. how likely the observation data is generated by the given process model. In most cases, as shown in Figure 1, we cannot observe the states and their transition activities of a dynamic process directly. Otherwise, it could be easy to detect such a process by evaluating the observation sequence against its state transition sequence. Instead, the evidence of the existence of such a process is often scattered over observation data that is tainted by noise. Therefore, filtering algorithms such as the Kalman Filter [1] for linear models and Viterbi algorithm [2] for Hidden Markov Models (HMM) are needed to correlate observation sequences and accumulate evidence for process detection.

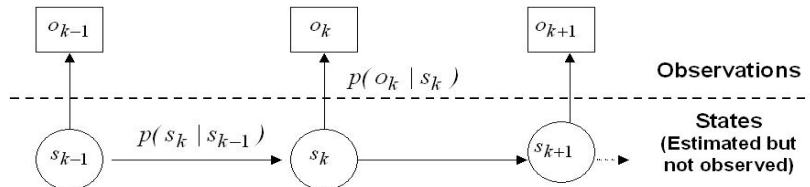


Figure 1: A dynamic process and its observations

Hidden Markov models were first introduced in speech recognition [2] and have been applied in computational biology [3] as well in recent years. In a HMM, state transitions follow a first-order Markov chain and an observable symbol is emitted according to some probability distribution each time a state is entered. We believe that HMMs can also be applied to characterize dynamic processes in defense and security applications. For example, Warrender and Forrest [4]

employed HMMs to characterize the system calls of computer software and further used these HMMs as the profile of software to detect anomaly behavior. One challenging problem for process detection (for other detection problems too) is how to build accurate process models in the first place. Mainly there are two approaches to build these models: One is to use expert knowledge to build models based on our understanding on a process; The other is to use data mining technology to learn models based on the training data of a process. The accuracy of a model is very much evaluated by trial and error in real applications. In many process detection problems, we don't have sufficient data to learn and evaluate a model so that it's usually hard to build an accurate model. For example, the intelligence data about terrorism activities is very rare but a lot of data is needed to statistically learn the accurate state transition probabilities and emission probabilities in a HMM. One alternative is to characterize the normal environment instead of the terrorism activities so that we can detect anomaly situations. Though we may have sufficient data to model a normal environment, anomaly detection is sensitive to normal changes of an environment and is not able to distinguish the type of anomaly activities. Moreover the characterization of an environment is much more complicated than the first case since we need to determine how abstract we should profile a normal environment. The abstraction level of process models has to balance its sensitivity to anomaly situations with its sensitivity to normal changes.

An inaccurate model could lead to high false alarm and misdetection rates in process detection and the inference result could be misleading in the decision-making process. To this end, we propose nonparametric weak process models derived from HMMs to describe abstract dynamic processes. In weak models, we don't specify state transition probabilities and emission probabilities but only the reachability between states and observations, i.e. the elements in state transition matrix and emission matrix are either one or zero. Therefore a weak model doesn't need the strong requirement for probability specification as in HMMs, which can dramatically reduce the difficulty and complexity in process modeling. Our principle is that if we are not able to build a precise model, it's better to build an abstract but accurate weak model rather than a fine but inaccurate model. In this paper, we analyze the properties of such weak models at first and then propose inference algorithms to compute the hypotheses of the hidden state sequence and the size of hypothesis set. Further, we analyze how we can control the size of hypothesis set by increasing the number of sensors in process detection.

2. HIDDEN MARKOV MODEL AND PROCESS DETECTION

There are several good tutorials on hidden Markov Models. Rabiner [2] gave a good introduction to the theory of hidden Markov models and their applications to speech recognition. Recently Ephraim and Merhav [5] gave an overview of statistical and information-theoretic aspects of hidden Markov models and introduced many new results developed in recent years. In this paper, we assume that readers already have the basic knowledge about a HMM and its theory. Otherwise they may want to read those tutorials first.

2.1 Hidden Markov model

As introduced in many HMM literatures, a HMM is characterized by the following parameters:

- 1.) N , the number of states in the model. Denote the individual states as $S = \{s_1, s_2, \dots, s_N\}$ and the state at time t as s^t . Denote the state sequence up to time t as $S^t = \{s^1, s^2, \dots, s^t\}$.
- 2.) M , the number of distinct observations in the model. Denote the individual observations as $O = \{o_1, o_2, \dots, o_M\}$ and the observation at time t as o^t . Denote the observation sequence up to time t as $O^t = \{o^1, o^2, \dots, o^t\}$.
- 3.) A , the matrix of state transition probability distribution in the model. $A = \{a_{ij}\}$ is the state transition matrix, where $A(s_i, s_j) = a_{ij} = p(s^{t+1} = s_j | s^t = s_i)$, $1 \leq i, j \leq N$.
- 4.) B , the matrix of emission probability distribution in the model. $B = \{b_{jk}\}$ is the emission matrix, where $B(s_j, o_k) = b_{jk} = p(o^t = o_k | s^t = s_j)$, $1 \leq j \leq N$ and $1 \leq k \leq M$.
- 5.) π , the initial state distribution. $\pi = \{\pi_i\}$, where $\pi_i = p\{s^1 = s_i\}$, $1 \leq i \leq N$.

For convenience, we denote a HMM with these parameters as $\lambda = \{A, B, \pi\}$.

2.2 Process detection problem

Assuming that several dynamic processes are characterized with HMMs λ_i ($1 \leq i \leq L$), the process detection problem here is to identify which process model is generating the incoming observation sequence O^t . Here we denote that the real process generating the observation sequence O^t as λ_0 . If the probabilities $p(\lambda_i|O^t)$ are known, we can compare these probabilities to determine the most likely process. According to the Bayesian rule, we have $p(\lambda_i|O^t) = p(O^t|\lambda_i)p(\lambda_i)/p(O^t)$. Now if we compare the probabilities $p(\lambda_i|O^t)$ of two models λ_i and λ_j , we have $\frac{p(\lambda_i|O^t)}{p(\lambda_j|O^t)} = \frac{p(O^t|\lambda_i)p(\lambda_i)}{p(O^t|\lambda_j)p(\lambda_j)}$. In most cases, probabilities $p(\lambda_i)$ and $p(\lambda_j)$ are unknown or difficult to be estimated so that in practice we often assume that they are equal. Therefore we can evaluate the ratio $p(O^t|\lambda_i)/p(O^t|\lambda_j)$ against a threshold to determine which process model matches the observations more closely.

Given a model λ_i , $p(O^t|\lambda_i)$ can be easily computed with the so-called Forward Procedure in HMM literatures [2][5]. One important issue for process detection is the detection accuracy. Since the analytical form of the probability distribution $p(O^t|\lambda_i)$ is usually unknown, we cannot use Neyman-Pearson detection theory [6] to conclude the related false alarm and misdetection rate from the selected threshold. However, with the following inequalities, we can use the Sequential Probability Ratio Test (SPRT) [7] to conclude the bounds for false alarm rate and misdetection rate,

$$F < r^t = \frac{p(O^t|\lambda_i)}{p(O^t|\lambda_j)} < G, \quad G \text{ and } F \text{ are two thresholds.} \quad (1)$$

The SPRT works in the following way: If the ratio r^t is bigger than G , we conclude that the process is λ_i . Conversely, if r^t is smaller than F , we conclude that the process is λ_j . If r^t is smaller than G but bigger than F , we continue to receive new observations until the ratio passes across these thresholds. Denote the false alarm rate as $\alpha = p_{\lambda_0=\lambda_j}(r^t > G)$, i.e. the real process is λ_j but the ratio r^t is bigger than G . Similarly denote the misdetection rate as $\beta = p_{\lambda_0=\lambda_i}(r^t < F)$. Based on the result of the sequential analysis [7], we can have the following relationship between false alarm rate, midsection rate and thresholds: $1 - \beta \geq G\alpha$ and $\beta \leq (1 - \alpha)F$.

The value of $p(O^t|\lambda_i)$ decreases exponentially with the growth of time t so that a better metric was proposed to measure how well the model matches the observations in [8]. Denote $H^t(\lambda_i) = -\frac{1}{t} \log p(O^t|\lambda_i)$. It was proved that $H^t(\lambda_i)$ converges to a constant value as $t \rightarrow \infty$, i.e. $\lim_{t \rightarrow \infty} H^t(\lambda_i) = H(\lambda_i)$. Moreover, we have $H(\lambda_0) \geq H(\lambda_i)$ for any models λ_i , i.e. the real process λ_0 generating the observation sequence O^t always has the maximal value $H(\lambda_0)$. For the process detection problem here, $H(\lambda_0)$ is unknown because we don't know what is λ_0 . In the above paragraph, we compare two models and estimate the detection accuracy under the assumption that the real process λ_0 is either λ_i or λ_j . In practice many process detection problems are not a binary detection problem and the real process λ_0 could be neither λ_i nor λ_j . Instead, given a process model λ_i and a sequence of observations O^t from sensors, we want to know how likely the real process is λ_i . For example, given a process description of terrorism activity, we want to know how likely it is this activity causes the intelligence data we observed. This problem is challenging because we don't know how many other processes can lead to the same observation sequence. However, we may use the Algorithm 2.1 to get a certain detection confidence:

Algorithm 2.1:

- 1.) Let a new model $\bar{\lambda}^0 = \lambda_i$, i.e. $\bar{\lambda}^0$ starts with the same parameters as in λ_i at time $t = 0$.
- 2.) With the observation sequence O^t , incrementally update the parameters of model $\bar{\lambda}^t$ as in the *Baum-Welch* method [2] or other gradient technologies such that $p(O^t | \bar{\lambda}^t) \geq p(O^t | \bar{\lambda}^{t-1})$.
- 3.) Compute and compare $H^t(\lambda_i)$ with $H^t(\bar{\lambda}^t)$ after a large t .

Straightforwardly, because $H(\lambda_0) \geq H(\lambda_i)$ for any models λ_i , if $H^t(\bar{\lambda}^t)$ is much bigger than $H^t(\lambda_i)$, we can conclude that λ_i is not likely to be the real process. However if the values of $H^t(\bar{\lambda}^t)$ and $H^t(\lambda_i)$ are close, we are not able to say that λ_i is likely to be the real process because those learning methods in Step 2 can only lead to a local maximum of $H^t(\bar{\lambda}^t)$. After a process is detected (the model λ_i is selected), the hidden state sequence of this process can be computed with the well-known Viterbi algorithm [2].

3. WEAK PROCESS MODELS

The accuracy of process detection strongly relies on the accuracy of those process models. The above process detection methods of HMMs work only if we can build accurate models for various dynamic processes in the first place. Many defense and security applications don't have sufficient training data for process modeling so that it is usually difficult to estimate those probabilities in HMMs. To this end, we derive weak models from HMMs to characterize dynamic processes. We believe that if we are not able to build a precise model, it is better to build a weak but accurate model rather than an inaccurate HMM. In the following sections, we will discuss the properties of such weak models and propose inference algorithms to compute the hypotheses of the hidden state sequence.

3.1 Weak model

Compared to HMM's mathematical formulation in section 2, a weak model has the following difference:

- 1.) A , the state transition matrix in the model. $A = \{a_{ij}\}$ and $A(s_i, s_j) = a_{ij} = 0 \text{ or } 1, 1 \leq i, j \leq N$. If the state could transfer from s_i to s_j in one discrete step, $a_{ij} = 1$; Otherwise $a_{ij} = 0$.
- 2.) B , the emission matrix in the model. $B = \{b_{jk}\}$ and $B(s_j, o_k) = b_{jk} = 0 \text{ or } 1, 1 \leq j \leq N \text{ and } 1 \leq k \leq M$. If the state s_j could emit the observation o_k , $b_{jk} = 1$; Otherwise $b_{jk} = 0$.

In weak models, we don't specify the state transition probabilities and emission probabilities as in HMMs but only the reachability between states and observations, i.e., we do not quantify the likelihood of state transition and observation emission. We believe that this abstraction can dramatically reduce the difficulty and complexity in process modeling. Figure 2 illustrates the difference of HMMs, Fuzzy weak models and Nonparametric weak models. In a Fuzzy weak model, the likelihood of state transition and observation emission is quantified with qualitative measures such as "unlikely", "likely" and "very likely". These fuzzy measurements could be useful in hypothesis ranking. In this paper, we focus our analysis on nonparametric weak models and our results can be easily upgraded for Fuzzy weak models by using the extra information of qualitative measures.

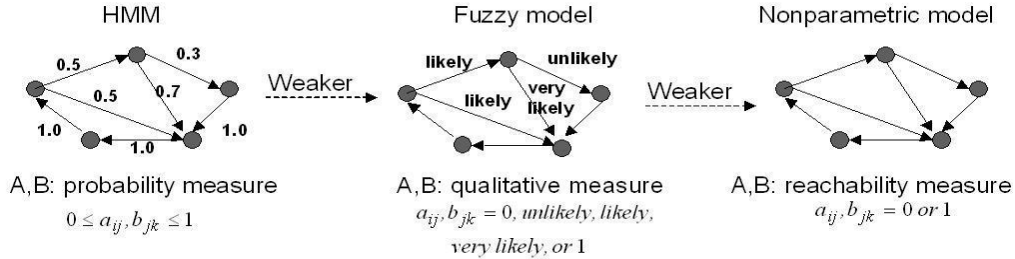


Figure 2: HMM, Fuzzy weak model and Nonparametric weak model

3.2 Basic problems

For weak models to be useful in process modeling and detection, we must solve several basic problems of interest in this section. There are similar problems for HMMs. However since there is no probability specification in weak models, we have to use different approaches to address these problems.

Problem 1: Given the observation sequence O^t and a weak model λ , how do we infer whether this model λ can generate the observation sequence O^t or not?

Problem 2: Given the observation sequence O^t and the weak model λ , how do we compute the corresponding state sequence S^t ?

Here we analyze the solution to the Problem 2 first because we can solve Problem 1 easily by analyzing the results from the Problem 2. Denote a hypothesis of the state sequence at time k as $\Omega_v^k = \{s^1, s^2, \dots, s^k\}$, where $1 \leq v \leq I^k$ and I^k is the total number of hypotheses at time k . Denote the hypothesis set at time k as $\Omega^k = \{\Omega_v^k\}$, $1 \leq v \leq I^k$. Given a weak model $\lambda = (A, B)$, we propose the following recursive algorithm to compute all hypotheses of the state sequence underlying the observation sequence O^t .

Algorithm 3.1:

1.) Initialization: at time step $t = 1$, with the observation o^1 , we have

$$\Omega_v^1 = \{s_i \mid B(s_i, o^1) = 1\}, 1 \leq i \leq N, 1 \leq v \leq I^1, I^1 = \sum_{i=1}^N B(s_i, o^1); \quad (2)$$

2.) Induction: at time $t = k + 1$, with the new observation o^{k+1} , we have

$$\Omega_g^{k+1} = \{(\Omega_v^k, s_j) \mid B(s_j, o^{k+1}) = 1, A(s^k, s_j) = 1, s^k \in \Omega_v^k\}, 1 \leq j \leq N, 1 \leq v \leq I^k, 1 \leq g \leq I^{k+1}, 1 \leq k \leq T - 1; \quad (3)$$

3.) Termination: at time $t = T$, we have I^T valid hypotheses:

$$\Omega_v^T, 1 \leq v \leq I^T.$$

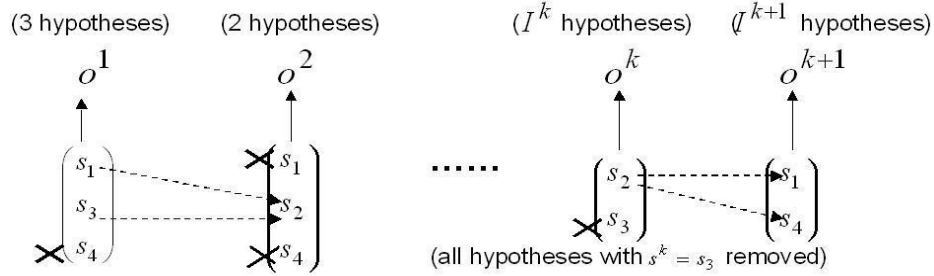


Figure 3: an example of the Algorithm 3.1

Figure 3 gives an example to illustrate how the Algorithm 3.1 works. Assume that we have a weak model $\lambda = (A, B)$ and

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, B = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \end{pmatrix}.$$

Assume $o^1 = o_2$ at time $t = 1$, according to the emission matrix B , three states $s^1 = \{s_1, s_3, s_4\}$ could emit o_2 , i.e. $B(s_i, o^1) = 1, i = 1, 3, 4$. Therefore at time $t = 1$ we have three hypotheses. Assume $o^2 = o_1$ at time $t = 2$, according to the matrix B , three states $s^2 = \{s_1, s_2, s_4\}$ could emit o_1 . As shown in Figure 3, according to the state transition matrix A , only two state transitions are possible from s^1 to s^2 , i.e. $A(s_1, s_2) = 1$ and $A(s_3, s_2) = 1$. That means s^1 cannot be

s_4 and s^2 cannot be s_1 or s_4 . At time $t = 2$, we have two hypotheses (s_1, s_2) and (s_3, s_2) . In fact in this case we are sure that $s^2 = s_2$ because both valid hypotheses have to pass the state s_2 . Similarly at time $t = k$, assume $o^k = o_4$ and two state $s^k = \{s_2, s_3\}$ could emit o_4 . At time $t = k + 1$, assume $o^{k+1} = o_3$ and two state $s^{k+1} = \{s_1, s_4\}$ could emit o_3 . According the transition matrix A , $s^k = s_3$ cannot transfer to the state s_1 or s_4 so that all early hypotheses with $s^k = s_3$ should be removed from the hypothesis set. Conversely, $s^k = s_2$ can transfer to both states s_1 and s_4 so that at time $t = k$ all hypotheses with $s^k = s_2$ are extended to two sets of new hypotheses with $s^{k+1} = s_1$ or $s^{k+1} = s_4$ at time $t = k + 1$.

Basically the Algorithm 3.1 concludes a set of valid hypotheses with regard to the constraints defined in structure of the matrix A and B . The Algorithm 3.1 is a forward procedure and works recursively even if we don't know the termination time T . If we know the whole observation sequence o^T , we can develop a similar backward procedure to conclude the hypotheses of the state sequence, i.e. starting from $t = T$ and terminating at $t = 1$. Due to its similarity to the Algorithm 3.1, we don't list the backward procedure here. Since we don't specify state transition probabilities and observation emission probabilities in weak models, the likelihood of each hypothesis cannot be ranked directly. In Fuzzy weak models, we can use the qualitative measures to evaluate the likelihood of each hypothesis. For example, we can count how many "very likely", "likely" and "unlikely" measures along the state sequence of each hypothesis and use these statistical numbers to roughly rank the likelihood of each hypothesis.

3.3 Efficient data structure for hypothesis representation

Given the observation sequence, the size of the hypothesis set is determined by the structure of the matrix A and B . In general the size of hypothesis set could be small if the matrix A and B are very sparse. With some structure of A and B , the Algorithm 3.1 may generate a large number of hypotheses. For example, if each element in A and B is one, i.e. $a_{ij} = 1$ and $b_{jl} = 1$ for $1 \leq i, j \leq N$ and $1 \leq l \leq M$, the size of hypothesis set $I^T = N^T$ and its growth is exponential along time T . Therefore we need an efficient data structure to represent the whole hypothesis set. Though the Algorithm 3.1 illustrates how to update each hypothesis independently, we don't have to maintain the state sequence of each hypothesis independently. Because of the Markov property of state transitions, we can use link lists to represent the whole hypothesis set. Denote a vector $Q^k = [q_1^k, q_2^k, \dots, q_N^k]^T$ where q_i^k represents the set of previous states that can transfer to the state $s^k = s_i$ in one step. $q_i^k = Null$ if $s_i \notin s^k$ ($B(s_i, o^k) = 0$) or $s^k = s_i$ doesn't have any previous state. For example, $q_i^k = \{s_1, s_2\}$ means that at time $t = k - 1$, the state $s^{k-1} = s_1$ or s_2 can transfer to the state $s^k = s_i$. As shown in Figure 3, $q_2^2 = \{s_1, s_3\}$ and $q_4^{k+1} = \{s_2\}$. The vectors Q^k can be computed with the following algorithm:

Algorithm 3.2:

1.) Initialization: at time step $t = 1$, with the observation o^1 , we have

$$q_i^1 = \left\{ s_i \mid B(s_i, o^1) = 1 \right\}, 1 \leq i \leq N; \quad (4)$$

2.) Induction: at time $t = k + 1$, with the new observation o^{k+1} , we have

$$q_j^{k+1} = \left\{ q_i^{k+1}, s_i \mid q_i^k \neq Null, B(s_j, o^{k+1}) = 1, A(s_i, s_j) = 1 \right\}, 1 \leq i, j \leq N, 1 \leq k \leq T - 1; \quad (5)$$

3.) Termination: at time $t = T$, we have T vectors Q^k ($k = 1, \dots, T$).

Therefore we can use T vectors Q^k ($k = T, T - 1, \dots, 1$) to represent the whole hypothesis set that could include as many as N^T hypotheses. At time $t = k$, following the vectors Q^k, Q^{k-1}, \dots, Q^1 , we can use a backward procedure to assemble all valid hypotheses up to time $t = k$. For example, as shown in Figure 3, $q_4^{k+1} = \{s_2\}$ and then go backward

one step to check q_2^k and so on. Similarly we can use the vector q_i^k to represent the set of next states instead of the previous states and develop a similar algorithm. The Algorithm 3.2 is more efficient because the backward procedure guarantees all hypotheses starting from $t = k$ can find valid state sequences backward to $t = 1$. Forward procedures like the Algorithm 3.1 have to remove invalid hypotheses during the iteration process. However, if the size of hypothesis set is small, the Algorithm 3.1 could be efficient because it maintains all valid hypotheses up to each step and doesn't need another backward procedure to assemble the hypotheses.

4. MULTI-ORDER PROPERTIES FOR PROCESS DETECTION

In fact we can view the first problem in section 3.2 as a detection problem. Given several weak models $\lambda_i (1 \leq i \leq L)$ and the observation sequence O^t , the solution to the Problem 1 allows us to conclude which process could cause the observed data. At termination time $t = T$, if the size of valid hypothesis set Ω^T is not zero, i.e. $I^T \neq 0$, we say that the weak model λ_i could generate the observation sequence O^T . That means the process associated with this weak model λ_i could be the origin of the observation sequence. Conversely, if the hypothesis set Ω^T is empty, we are sure that the process model λ_i is not the origin of the observed data. If we only want to determine the size of hypothesis set but not the state sequence, the Algorithm 5.1 in the following section can solve the Problem 1 efficiently. If several models could cause the same observation sequence, we may need other evidence to distinguish these models.

We believe that the hypothesis set can be analyzed to extract other hidden information about the process. For each hypothesis at time $t = k$, we use a vector $W^k = (w_1^k, w_2^k, \dots, w_N^k)$ to record how many times the state sequence of this hypothesis has traversed each state. w_i^k is the number of times that this state sequence has traversed the state s_i up to time $t = k$. We can view this vector as an inherent property of each hypothesis and it can be recursively updated during the computing process of the Algorithm 3.1. If a hypothesis is removed at time $t = k + 1$, the vector of this hypothesis is abandoned. Otherwise assume that the new hypothesis $\Omega_v^{k+1} = (\Omega_i^k, s_j)$, the vector $W^{k+1} = (w_1^k, \dots, w_j^k + 1, \dots, w_N^k)$. At the termination time T , we have I^T valid hypotheses and I^T vectors $W_v^T (1 \leq v \leq I^T)$. If we analyze these I^T vectors, at least we can conclude:

- 1.) whether all these hypotheses traversed certain states. In fact by comparing these vectors, we know the minimal and maximal number of visits for each state. In some detection problems, some states could be the "high-risk" state and an alert should be generated if we are sure that these states were traversed based on the observation. The total visit number or frequency of each state could also be used in some detection problems.
- 2.) whether all these hypotheses didn't traverse certain states. This is a dual problem of the above problem.

These are just two examples of the hidden information that we can extract from these vectors. For different problems, other useful information could be extracted from these vectors. For example, we may be able to develop a function of these vectors $f(W_v^k)$ and use this function value to rank the likelihood of hypotheses in some problems. Instead of using a vector to record the number of state visits, we can also use a matrix $Q^k = \{q_{ij}^k\}_{N \times N}$ to record the state transition history for each hypothesis. q_{ij}^k is the number of times that the state s_i transferred to the state s_j up to time $t = k$. Q^k can be recursively computed in the same way as W^k . W^k is a vector and Q^k is a matrix. They are both inherent properties of each hypothesis so that we name W^k and Q^k as first-order and second-order properties of hypotheses, respectively. Similarly, by comparing I^T matrices Q_v^k , we can extract other hidden information about state transitions. The state transition matrix A can be represented with a directed graph, where states are the nodes and state transitions are the edges. With second-order properties Q_v^k , we can determine whether all these hypotheses traversed certain edges

or not and this information could also be used in some detection problems. Theoretically, we can develop higher order properties to analyze hypotheses if it's necessary for some detection problems.

These properties of hypotheses could also be used to distinguish processes in detection. For example, assume that we have two weak models λ_1 and λ_2 . At the termination time T , all hypotheses of the weak model λ_1 visited a specific state that represents a certain event in a real detection problem. Conversely all hypotheses of the weak model λ_2 didn't visit any state that represents that event. Based on other information source, if we know that event did happen, we can be sure that the real process is λ_1 but not λ_2 . Each hypothesis has its property and each process has a set of properties of its hypotheses. Statistical analysis on the multi-order properties of hypotheses could identify the unique character of a dynamic process.

5. SIZE OF HYPOTHESIS SET AND COMPUTING COMPLEXITY

As discussed earlier, for each hypothesis, these multi-order properties can be recursively computed with polynomial time. However, with an observation sequence O^t , a weak model may have a large number of hypotheses resulting from the Algorithm 3.1 or 3.2 and the size of hypothesis set may grow exponentially. Therefore the total computing complexity to analyze the properties of the hypothesis set could grow exponentially too. In this section, we propose an algorithm to compute the size of hypothesis set for weak models. The computing complexity of hypothesis analysis is roughly proportional to the size of the hypothesis set.

As in section 3.2, we denote I^k as the total number of hypotheses at time k . Further we denote I_i^k ($1 \leq i \leq N$) as the number of hypotheses with the last state $s^k = s_i$ and we have $I^k = \sum_{i=1}^N I_i^k$. For convenience, use $b_j(o^k)$ to represent the element $B(s_j, o^k)$ in the emission matrix B . Given a weak model $\lambda = (A, B)$ and an observation sequence O^t , we propose the following recursive algorithm to calculate the size of hypothesis set.

Algorithm 5.1:

- 1.) Initialization: at time $t = 1$, with the observation o^1 , we have

$$I_i^1 = B(s_i, o^1), 1 \leq i \leq N; \quad (6)$$

- 2.) Induction: at time $t = k + 1$, with the new observation o^{k+1} , we have

$$I_j^{k+1} = \left[\sum_{i=1}^N (I_i^k \cdot a_{ij}) \right] b_j(o^{k+1}), 1 \leq j \leq N, 1 \leq k \leq T - 1; I^k = \sum_{j=1}^N I_j^k; \quad (7)$$

- 3.) Termination: at time $t = T$, we have I^T valid hypotheses,

$$I^T = \sum_{j=1}^N I_j^T. \quad (8)$$

The size of hypothesis set can be recursively calculated with the Algorithm 5.1 because the elements in the matrix A and B are either one or zero. This algorithm is an efficient solution to the Problem 1 in section 3.2 if we don't need to know the state sequence. The Algorithm 5.1 is a forward procedure and we can develop a similar backward procedure starting from $t = T$ to compute the size of hypothesis set. Denote a vector $R^k = (r_1^k, r_2^k, \dots, r_N^k)$ and r_i^k represents the set size of q_i^k in section 3.3. Based on these vectors R^k , another backward procedure can be developed to compute the size of hypothesis set if the historical part of the observation sequence is not saved.

In some applications, we may only want to keep the most recent L steps of the state sequence but not the whole state sequence, i.e. at time $t = k$ ($k > L$), all hypotheses should ignore any difference of their state sequences before

$t = k - L + 1$. For the Algorithm 3.2, it is very efficient to update the L -step hypothesis set by stopping the backward procedure at $t = k - L + 1$ instead of $t = 1$. For the Algorithm 3.1, there are two approaches to update the hypothesis set:

- 1.) For time step $t = k (k > L)$, after the step 2 of the Algorithm 3.1, we can run a procedure to prune the hypotheses. The prune procedure is to cut the early part ($t \leq k - L$) of the state sequence and keep the state sequence starting from $t = k - L + 1$ to $t = k$ for each hypothesis. Then compare the remaining part of hypotheses and abandon the duplicate ones.
- 2.) For time step $t = k (k > L)$, reset the hypothesis set at time $t = k - L + 1$ with the following equation:

$$\Omega_v^{k-L+1} = \left\{ \mathcal{S}_i \mid I_i^{k-L+1} > 0 \right\}, 1 \leq i \leq N, 1 \leq v \leq I^{k-L+1}. \quad (9)$$

Then run the step 2 of the Algorithm 3.1 for L steps to generate the new hypothesis set at time $t = k$.

If the size of hypothesis set is small and the shifting time window L is big, the first procedure is more efficient than the second one. Otherwise the second procedure is more efficient. In some case, we don't need to keep the size of the time window strictly. As long as the size of hypothesis set is manageable, we can keep the whole state sequence. Periodically, the above procedures can be applied to prune hypothesis set after the size of hypothesis set is bigger than a selected threshold. These problems don't exist in the Algorithm 3.2 so that it's better to use Algorithm 3.2 while the time window is shifting.

6. STRUCTURE OF EMISSION MATRIX

Given a sequence of observations O^t , the size of hypothesis set is determined by the structure of the matrix A and B . The structure of the state transition matrix A is determined by the physical constraints of a dynamic process and cannot be altered in detection. However, we can change the structure of the emission matrix B if more sensors are added to observe the states in that dynamic process. In this section we analyze how to control the size of hypothesis set by adding sensors and tuning the structure of the emission matrix.

6.1 Stability analysis

Define a vector $\Phi^k = (I_1^k, I_2^k, \dots, I_N^k)^T$. The Equation (7) can be rewritten with the following format:

$$\Phi^k = \begin{pmatrix} I_1^k \\ I_2^k \\ \vdots \\ I_N^k \end{pmatrix} = \begin{bmatrix} b_1(o^k) & 0 & \dots & 0 \\ 0 & b_2(o^k) & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & b_N(o^k) \end{bmatrix} \begin{bmatrix} a_{11} & a_{21} & \dots & a_{N1} \\ a_{12} & a_{22} & \dots & a_{N2} \\ \dots & \dots & \dots & \dots \\ a_{1N} & a_{2N} & \dots & a_{NN} \end{bmatrix} \begin{pmatrix} I_1^{k-1} \\ I_2^{k-1} \\ \dots \\ I_N^{k-1} \end{pmatrix} = D(o^k) A^T \Phi^{k-1} \quad (10)$$

where $D(o^k)$ is the diagonal matrix in the Equation (10) and $D(o^k) \in \{D(o_1), D(o_2), \dots, D(o_M)\}$. For convenience, denote $D^k = D(o^k)$. The form of the Equation (10) is usually viewed as a linear, time-variant and zero input system in

control theory literature. According to the Norm-1 definition, we have $\|\Phi^k\|_1 = \sum_{i=1}^N I_i^k = I^k$. The question is: Given the

state transition matrix A of a dynamic process, can we choose a structure of the emission matrix B to make $\|\Phi^k\|_1$ not

grow exponentially? This is critical for weak models because an exponential growth of the size of hypothesis set is not acceptable in practice. Given any structure of the state transition matrix A , we can always find an emission matrix so that the size of hypothesis set will not grow exponentially under any sequence of observations. For example, each

column of the matrix B only has one element with value one, i.e. $\sum_{i=1}^N b_i(o_j) = 1$ for any $1 \leq j \leq M$. With this emission

matrix, the observation sequence can be mapped to the state sequence deterministically and we only have one hypothesis - the real process. In practice we may need a large number of sensors to achieve that requirement and it is also unnecessary for some structure of A . The challenging question is: Given a specific structure of A , what is the necessity condition of the structure of B to make the size of hypothesis set not grow exponentially?

Lets consider the worst case: Each element of the matrix A is one, i.e. any state can transfer to another state and the structure of A doesn't place any constraints on state transitions. Multiply a vector $[1,1,\dots,1]_N$ to each side of the Equation (10) and we have:

$$I^k = \sum_{i=1}^N I_i^k = \begin{bmatrix} b_1(o^k) & b_2(o^k) & \dots & b_N(o^k) \end{bmatrix} \begin{bmatrix} I^{k-1} & I^{k-1} & \dots & I^{k-1} \end{bmatrix}_N^T = \sum_{i=1}^N b_i(o^k) I^{k-1} \quad (11)$$

Denote $b_{min} = \min_{1 \leq j \leq N} \sum_{i=1}^N b_i(o_j)$ and $b_{max} = \max_{1 \leq j \leq N} \sum_{i=1}^N b_i(o_j)$, we have $1 \leq b_{max}, b_{min} \leq N$ and

$$(b_{min})^{k-1} I_1 \leq I^k \leq (b_{max})^{k-1} I_1 \quad (12)$$

From inequalities (12), we know that the size of hypothesis set will not grow exponentially only if $b_{max} = 1$. This is a necessity condition for the structure B , given the structure of A whose elements are all one.

Lets consider another simple case: Assume that each column of the matrix B is same, i.e. the observations are useless to distinguish any state and the deployed sensors are totally blind in sensing the states of a dynamic process. In this case, $D(o_i)$ is same for $1 \leq i \leq N$ and denote $D = D(o_i)$. The Equation (10) can be viewed as a time-invariant linear system and it can be written as $\Phi^k = (DA^T)^{k-1} \Phi^1$. With this equation, it is well known that the stability of Φ^k is determined by the eigen values of the matrix DA^T . If DA^T has N distinct real eigen values $\lambda_i (1 \leq i \leq N)$ and $\max_i |\lambda_i| < 1$, Φ^k asymptotically decreases. If DA^T has other forms of eigen values, see the specific stability conditions in [9].

In most case, the emission matrix B has different columns in order to distinguish states. We can have the following equations by using the Equation (10) recursively:

$$\Phi^k = \left[\prod_{t=2}^k (D^t A^T) \right] \Phi^1 \quad (13)$$

$$\|\Phi^k\| = \left\| \prod_{t=2}^k (D^t A^T) \Phi^1 \right\| \leq \left\| \prod_{t=2}^k D^t A^T \right\| \|\Phi^1\| \quad (14)$$

For any possible sequence $(D_k, D_{k-1}, \dots, D_2)$, if these exists a sufficient large k such that $\left\| \prod_{t=2}^k D^t A^T \right\| < 1$, the Equation (14) is a contraction mapping and $\|\Phi^k\| (\|\Phi^k\|_1 = I^k)$ asymptotically decreases. Note that $D^k = D(o^k)$ is always one of the M matrices $\{D(o_1), D(o_2), \dots, D(o_M)\}$ and it is also dependent on $D^{k-1} = D(o^{k-1})$ since a given o^{k-1} can only transfer to a set of o^k . See the similar conditions for stability in [10][11][12]. However, this condition is very strong and it's not clear what the structure of B should be to satisfy this condition. We will analyze the necessity condition for stability and the necessary structure of the emission matrix to meet this condition in our future work.

6.2 Emission matrix and sensors

Assume that we have different types of sensors capable of sensing the states of a dynamic process and these sensors are binary sensors, i.e. these sensors output "1" or "0" to report whether they sense some events or not. Given the necessity condition of the structure of B , what is the minimal number of sensors needed to build that structure? Table 1 illustrates the correlation matrix between the states of a dynamic process and the sensors. Assume that a dynamic process has five states and two binary sensors are used to observe this process. These two binary sensors can distinguish $2^2 = 4$ observations. Let $o_1 = (0,0)$, $o_2 = (1,0)$, $o_3 = (0,1)$ and $o_4(1,1)$. Based on the correlation matrix in the table 1, we have the emission matrix $B = \{(1,0,0,0), (0,1,0,0), (0,0,1,0), (0,0,0,1), (0,0,1,0)\}$. This is how an emission matrix is formulated in practice. Straightforwardly, in order to observe each state directly, we need at least $\log_2 N$ different sensors. Meanwhile, if we increase the number of sensors, we can make the emission matrix sparser, which eventually can reduce the size of the hypothesis set in the reasoning process.

Table 1: The correlation matrix between states and sensors

	State 1	State 2	State 3	State 4	State 5
Sensor 1	0	1	0	1	0
Sensor 2	0	0	1	1	1

Given the necessity condition of the structure of B , we assume that this matrix B is a $N \times M$ dimension matrix. Since the Hamming distance of any two columns of the matrix B is at least one (otherwise two observations should be merged as one), we need at least $\log_2 M$ unique sensors to distinguish these M observations. We need more sensors if the Hamming distances of pairs of these columns are bigger, which increases classification redundancy in a noisy environment.

7. CONCLUSIONS

Hidden Markov models are widely used in modeling dynamic processes. For many process detection problems in defense and security applications, we don't have sufficient training data to estimate the accurate probabilities in hidden Markov models. In this paper, we proposed a series of weak models to characterize dynamic processes. In weak models, we don't need the strong requirement for probability specification as in HMMs, which can dramatically reduce the difficulty and complexity in modeling dynamic processes. We analyzed the properties of such weak models and proposed recursive algorithms to compute the hypotheses of the state sequence and the size of the hypothesis set. Further we analyzed how to control the size of hypothesis set by increasing the number of sensors and tuning the structure of the emission matrix. In our future work, we will apply weak models in homeland security applications and further analyze the properties of weak models for robust process detection.

ACKNOWLEDGEMENTS

This work was partially supported by: ARDA Grant F30602-03-C-0248, DARPA projects F30602-00-2-0585 and F30602-98-2-0107, and Award No. 2000-DT-CX-K001 from the Office for Domestic Preparedness, U.S. Department of Homeland Security. Points of view in this document are those of the author(s) and do not necessarily represent the official position of the sponsoring agencies or the U.S. Government. Many thanks to Professor George Cybenko for his valuable inputs on this work.

REFERENCES

1. R.E. Kalman, "A new approach to linear filtering and prediction problems," *J. Basic Eng.*, vol. 82-D, 1969.
2. Lawrence R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition", *Proceedings of The IEEE*, vol. 77, no. 2, February 1989.
3. S.R. Eddy, "Hidden markov models", *Current Opinion in Structure Biology*, vol. 6.
4. B.P.C. Warrender and S. Forrest, "Detecting intrusion using system calls: alternative data models", *1999 IEEE Symposium on Security and Privacy*, 1999.
5. Y. Ephraim and N. Merhav, "Hidden Markov processes", *IEEE Transactions on Information Theory*, vol. 48, no. 6, 2002.
6. V. H. Poor, *An Introduction to Signal Detection and Estimation*, Springer-Verlag, 1994.
7. A. Wald, *Sequential Analysis*, John Wiley & Sons, 1947.
8. B.H. Juang and L.R. Rabiner, "A probabilistic distance measure for hidden Markov models", *AT&T Tech J.*, vol. 64, no.2, 1985.
9. O. Galor, *Introduction to stability analysis of discrete dynamical system*, Monograph in preparation, http://www.econ.brown.edu/fac/Oded_Galor/.
10. M. Sichitiu and P. Bauer, "Stability of discrete time-variant linear delay systems and applications to network control", *Proc. of the International IEEE Conference on Electronics, Circuits, and Systems (ICECS 2001)*, pp.985--989, Sep. 2001.
11. P. Bauer, K. Premaratne, and J. Duran, "A necessary and sufficient condition for robust asymptotic stability of time-variant discrete systems", *IEEE Trans. On Automatic Control*, vol. 38, pp.1427-1430, 1993.
12. A. Bhaya and F. Mota, "Equivalence of stability concepts for discrete time-varying systems", *International Journal of Robust and Nonlinear Control*, vol. 4, pp.725-740, 1994.