INTERNATIONAL JOURNAL OF INTELLIGENT CONTROL AND SYSTEMS
VOL. 10, NO. 1, MARCH 2005, 33-42

Robust Process Detection Using Nonparametric Weak Models

Guofei JIANG

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 is usually hard to estimate those state transition and emission probabilities precisely in practice, especially if the training data is not sufficient and the process is not stationary. To this end, we propose nonparametric weak models derived from HMMs to characterize dynamic processes. A weak model does not need the strong requirement for probability specification as in HMMs and it can also characterize non-stationary processes. 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. Furthermore, we analyze how to reduce the size of the hypothesis set by tuning the structure of the emission matrix.**

***Index Terms*— Process detection, inference algorithm, hidden Markov model, nonparametric model, and hypothesis**

1. INTRODUCTION

Many defense and security applications involve the detection of a dynamic process in large amount of observation data. For example, the detection of a moving target in sensor data, the detection of terrorism activity in intelligence data, and the detection of a multi-stage attack in computer logs. We believe that many of these dynamic activities can be described as a deterministic or stochastic dynamic process. A process model 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; terrorism activity could be described with a Markov model; a multi-stage computer attack could be represented in a finite state machine. Given a process model, the process detection problem is to identify the existence of such a process in large amount of observation data, i.e., how likely the observation data is generated by the given process model [5]. In some cases, as shown in Fig. 1, we cannot observe the states and their transition activities of a dynamic process directly. Otherwise, we can detect such a process by comparing 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 are needed to correlate observation sequences and accumulate evidence for process detection. Kalman Filter [11] for linear models and Viterbi algorithm [13] for Hidden Markov Models (HMM) are typical examples of such algorithms.

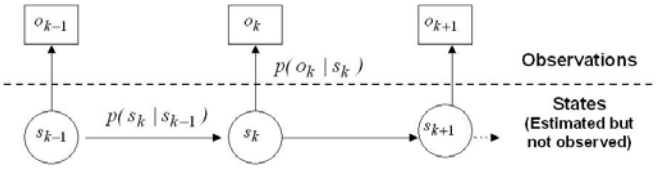


Fig. 1. A dynamic process and its observations.

HMMs were first introduced in speech recognition [13] and applied in computational biology [6] 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 when a state is entered. HMMs have also been applied to characterize dynamic processes in security applications. For example, Warrender and Forrest [16] employed HMMs to characterize the system calls of computer software and further used these HMMs as the profile of software to detect abnormal behavior. One challenging problem for process detection (for other detection problems too) is how to build accurate process models in the first place. There are two basic approaches to build these models: one is to use expert knowledge to build models based on our understanding of a process; the other is to use data mining technology to learn a model based on the training data of a process. The quality of a model is usually evaluated later by trial and error in real applications.

In many detection problems, we may not have sufficient training data that can be used to learn and evaluate a process model. For example, we do not have much data to characterize rare terrorist activity. Meantime, some real processes even may not be stationary. In these cases, it is difficult to learn an accurate HMM to characterize a process. Based on an inaccurate model, the detection result could be very misleading for decision-making. To this end, we propose nonparametric weak process models derived from HMMs to describe a class of dynamic processes. In weak models, we do not specify state transition probabilities and emission probabilities. Instead, we describe only the reachability between states and observations, i.e., the elements in state transition matrix and emission matrix are either one or zero. Compared to HMMs, weak models can reduce the difficulty and complexity of process modeling and are also not sensitive to parameter estimation errors. In this paper, we analyze the properties of such weak models at first and then propose inference algorithms based on dynamic programming to compute the hypotheses of the hidden state sequence and the size of hypothesis set.

Manuscript received September 20, 2004; revised February 4, 2005. This paper is extended from "Weak process models for robust process detection" published at Proc. of the SPIE 5403, Orlando, FL, USA, April, 2004.

G. Jiang is with NEC Laboratories America, Princeton, NJ 08540. This work was performed when the author worked at the Institute for Security Technology Studies (ISTS), Dartmouth College, Hanover, NH 03755. (His email: gfj@nec-labs.com).

Furthermore, we analyze how we can reduce the size of hypothesis set by tuning the structure of the emission matrix in process detection.

2. HMM AND PROCESS DETECTION

In this section, we introduce the process detection problems with regard to HMMs. There are several good tutorials on HMMs. Rabiner [13] gave a good introduction to the theory of HMMs and their applications to speech recognition. Recently Ephraim and Merhav [7] gave an overview of statistical and information-theoretic aspects of HMMs and introduced many new results developed in recent years. In this paper, we assume that readers already have the basic knowledge about HMM and its theory.

2.1 Hidden Markov model

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

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

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 ($1 \le i \le L$), the process detection problem here is to identify which process model is generating the incoming observation sequence O^t. Here we denote the real process generating the observation sequence O^t by λ_0. If the probabilities $p(\lambda_i \mid O^t)$ are known, we can compare these probabilities to determine the most likely process. According to the Bayesian rule, we have $p(\lambda_i \mid O^t) = p(O^t \mid \lambda_i) p(\lambda_i) / p(O^t)$. Now if we compare the probabilities $p(\lambda_i \mid O^t)$ of two models λ_i and λ_j, we have

$$\frac{p(\lambda_i \mid O^t)}{p(\lambda_j \mid O^t)} = \frac{p(O^t \mid \lambda_i) p(\lambda_i)}{p(O^t \mid \lambda_j) p(\lambda_j)}. \quad (1)$$

In most cases, probabilities $p(\lambda_i)$ and $p(\lambda_j)$ are unknown or difficult to estimate so that in practice we often assume that they are equal. Therefore we can evaluate the ratio $p(O^t \mid \lambda_i) / p(O^t \mid \lambda_j)$ against a threshold to determine which process model matches the observation data more closely.

Given a model λ_i, $p(O^t \mid \lambda_i)$ can be easily computed with the so-called Forward Procedure in HMM literature [13][7]. Since the analytical form of the probability distribution $p(O^t \mid \lambda_i)$ is usually unknown, we cannot use Neyman-Pearson detection theory [12] to conclude the related false alarm and misdetection rate from the selected threshold of Equation (1). However, with the following inequalities, we can use the Sequential Probability Ratio Test (SPRT) [15] to conclude the bounds of false alarm and misdetection rate,

$$F < r^t = \frac{p(O^t \mid \lambda_i)}{p(O^t \mid \lambda_j)} < G, \quad (2)$$

where G and F are two thresholds. The SPRT works in the following way: If the ratio r^t is bigger than G, we conclude that the observed process is λ_i. Conversely, if r^t is smaller than F, we conclude that the observed 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 one of these two 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 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 [15], we can have the following relationship between false alarm rate α, misdetection rate β and these two thresholds G and F: $1 - \beta \ge G\alpha$ and $\beta \le (1 - \alpha)F$.

The value of $p(O^t \mid \lambda_i)$ decreases exponentially with the growth of time t. Therefore a better metric was proposed in [10] to measure how well a model matches the observation. Denote $H^t(\lambda_i) = -\frac{1}{t} \log p(O^t \mid \lambda_i)$. It was proved that $H^t(\lambda_i)$ converges to a constant value as $t \to \infty$, i.e., $\lim_{t \to \infty} H^t(\lambda_i) = H(\lambda_i)$. Moreover, we have $H(\lambda_0) \ge H(\lambda_i)$ 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 the real process λ_0 is unknown. 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 such a binary classification 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. This problem is challenging because we do not know how many other processes can lead to the same observation sequence. However, we may use Algorithm 2.1 to obtain a certain level of detection confidence:

Algorithm 2.1:

1) Let a new model $\overline{\lambda}^0 = \lambda_i$, i.e. $\overline{\lambda}^0$ 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 $\overline{\lambda}^t$ as in the *Baum-Welch* method [3] or other gradient techniques so that we can have $p(O^t|\overline{\lambda}^t) \geq p(O^t|\overline{\lambda}^{t-1})$.
3) Compute and compare $H^t(\lambda_i)$ with $H^t(\overline{\lambda}^t)$ after a large time t.

Straightforwardly, since we have $H(\lambda_0) \geq H(\lambda_i)$ for any model λ_i, if $H^t(\overline{\lambda}^t)$ 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 $H^t(\overline{\lambda}^t)$ 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 mentioned in Step 2 can only lead to a local maximum of $H^t(\overline{\lambda}^t)$. 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 [13].

3. WEAK PROCESS MODELS

The accuracy of process detection strongly relies on the accuracy of the given process models. The above process detection method of HMMs works only if we can construct accurate models for various dynamic processes in the first place. Many defense and security applications do not have much training data for process modeling so that it is usually difficult to precisely estimate those probabilities in HMMs. In fact, without sufficient training data, the learned probabilities can hardly capture the stochastic property of a real process. We believe that if we are not able to build a precise model, it is better to use a weak but accurate model to characterize a dynamic process. Otherwise, the detection results based on an inaccurate model are meaningless and also misleading for decision making.

3.1 Weak models

Compared to HMM's mathematical formulation in Section 2, a weak model can be described in the following way:

1) A, the state transition matrix in the model. $A = \{a_{ij}\}$ and $A(s_i, s_j) = a_{ij} = 0$ or 1, $1 \leq i, j \leq N$. If the state s_i could transfer to the state s_j in one discrete time step (i.e., as long as $p(s^{t+1} = s_j | s^t = s_i) > 0$), $a_{ij} = 1$; otherwise $a_{ij} = 0$.
2) B, the emission matrix in the model. $B = \{b_{jk}\}$ and $B(s_j, o_k) = b_{jk} = 0$ or 1, $1 \leq j \leq N$ and $1 \leq k \leq M$. If the state s_j could emit the observation o_k (i.e., as long as $p(o^t = o_k | s^t = s_j) > 0$), $b_{jk} = 1$; otherwise $b_{jk} = 0$.

In weak models, we do not 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 reduce the difficulty and complexity of process modeling.

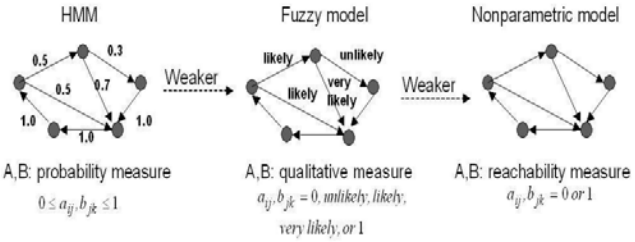


Fig. 2. HMM, fuzzy model and nonparametric model

Fig. 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 extended to fuzzy weak models by using the extra information of these qualitative measures. Weak models are also not as sensitive to parameter estimation errors as HMMs. For example, without accurate probability estimation in HMMs, the hypothesis ranking of hidden state sequences such as maximum likelihood is meaningless in Viterbi algorithm. Meantime, many real processes may not be stationary. The probabilities of a HMM could vary along the time though the reachability structure of the model is not changed. For example, the road network of a city seldom changes but the distribution of traffic in this network is always changing. In these cases,

weak models might be the only choice to characterize such processes.

3.2 Basic problems

For weak models to be useful in process modeling and detection, we must analyze the mathematical properties of weak models and solve the following basic problems. 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 hidden state sequence S^t?

Here we analyze Problem 2 first because we can solve the Problem 1 easily by analyzing the results of Problem 2. Denote a hypothesis of the hidden state sequence at time k as $\Omega_v^k = \{(s^1, s^2, ..., s^k)\}$, where $1 \le v \le I^k$ and I^k is the total number of hypotheses at time k. Denote the hypothesis set at time k as $\Omega^k = \{\Omega_v^k\}$, $1 \le v \le I^k$. 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

$$\Omega_v^1 = \{s_i \mid B(s_i, o^1) = 1\}, 1 \le i \le N,$$
$$1 \le v \le I^1,\ I^1 = \sum_{i=1}^{N} B(s_i, o^1); \quad (3)$$

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

$$\Omega_g^{k+1} = \{(\Omega_v^k, s_j) \mid B(s_j, o^{k+1}) = 1,\ A(s^k, s_j) = 1,\ s^k \in \Omega_v^k\},$$
$$1 \le j \le N,\ 1 \le v \le I^k,\ 1 \le g \le I^{k+1},\ 1 \le k \le T-1; \quad (4)$$

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

Ω_v^T, $1 \le v \le I^T$.

Fig. 3 gives an example to illustrate how 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 Fig. 3, according to the state transition matrix A, only two state transitions from s^1 to s^2 are possible, 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. Therefore at time $t = 2$, we only 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 to the transition matrix A, $s^k = s_3$ cannot transfer to the state s_1 or s_4 so that all early hypotheses ending 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 ending with $s^{k+1} = s_1$ or $s^{k+1} = s_4$ at time $t = k+1$.

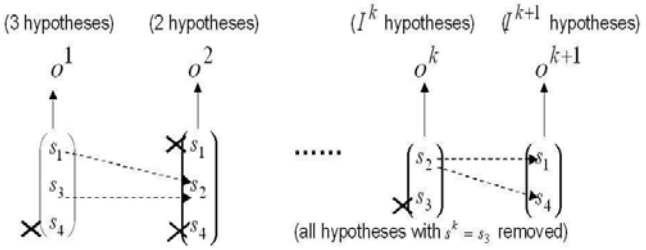


Fig. 3. An example of Algorithm 3.1.

Basically Algorithm 3.1 concludes a set of valid hypotheses with regard to the constraints defined in the structure of matrices A and B. Algorithm 3.1 is a forward procedure and works recursively even if we do not 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 Algorithm 3.1, we omit the backward procedure here. Since we do not specify state transition probabilities and observation emission probabilities in weak models, the likelihood of each hypothesis cannot be ranked directly. However, in fuzzy weak models, we can use the qualitative measures to evaluate the likelihood of each hypothesis. For example, we can count the number of "very likely", "likely" and "unlikely" measures along the state sequence of each hypothesis and use these statistical numbers to rank the likelihood of each hypothesis roughly.

3.3 Efficient data structure

Given the observation sequence O^T, the size of the hypothesis set is determined by A and B. In general the size of hypothesis set could be small if A and B are very sparse. We believe that many security applications may have sparse A and B. Conversely with some structure of A and B, 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 $1\le i,j\le N$ and $1\le l\le M$, 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 Algorithm 3.1 updates each hypothesis independently, we do not 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 $Q^k=\left[q_1^k,q_2^k,\cdots,q_N^k\right]^T$ where q_i^k represents the set of previous states that can transfer to the current state $s^k=s_i$ in one time step. $q_i^k=Null$ if $s_i\notin s^k$ ($B(s_i,o^k)=0$) or $s^k=s_i$ does not 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 Fig. 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
$$q_i^1=\left\{s_i\middle|B(s_i,o^1)=1\right\},\ 1\le i\le N; \tag{5}$$
2) Induction: at time $t=k+1$, with the new observation o^{k+1}, we have
$$q_j^{k+1}=\left\{\left(q_j^{k+1},s_i\right)\middle|q_i^k\ne Null,B(s_j,o^{k+1})=1,A(s_i,s_j)=1\right\},$$
$$1\le i,j\le N,\ 1\le k\le T-1; \tag{6}$$
3) Termination: at time $t=T$, we have T vectors $Q^k(k=1,\cdots,T)$.

Therefore we can use T vectors $Q^k(k=T,T-1,\cdots,1)$ to represent the whole hypothesis set that could include as many as N^T hypotheses. At time $t=k$, following the vectors $Q^k,Q^{k-1},\cdots$, and Q^1, we can use a backward procedure to assemble all valid hypotheses up to time $t=k$. For example, as shown in Fig. 3, we have $q_4^{k+1}=\{s_2\}$ 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. 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 Algorithm 3.1 have to remove invalid hypotheses during the iteration process. However, if the size of hypothesis set is small, Algorithm 3.1 could be efficient because it maintains all valid hypotheses up to each step and does not need another backward procedure to assemble the hypotheses. Both Algorithms 3.1 and 3.2 compute all possible hypotheses of hidden state sequences so that weak models are not as sensitive to parameter estimation errors as HMMs. If the size of hypothesis set is not large, we may want to analyze every hypothesis. This is necessary for some mission critical applications such as terrorist activity analysis because any misdetection could cause catastrophic disasters. The state sequence with maximum likelihood in HMMs only means that this state sequence is most likely with regard to the received observations but it does not have to be the real state sequence.

In fact we can view Problem 1 as a detection problem. Given several weak models λ_i ($1\le i\le L$) and an observation sequence O^t, the result of Problem 2 allows us to easily conclude which process could generate the observed data. At termination time $t=T$, if the size of valid hypothesis set Ω^T is not zero, i.e. $I^T\ne 0$, we say that the weak model λ_i could generate the observation sequence O^T. 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 hidden state sequence, Algorithm 5.1 proposed in Section 5 can solve Problem 1 more effectively. If several models could generate the same observation sequence, we may need other evidence to distinguish these models.

4. MULTI-ORDER PROPERTIES

The hypothesis set can be analyzed to extract other hidden information about the process. For each hypothesis at time $t=k$, here we use a vector $W^k=(w_1^k,w_2^k,\cdots,w_N^k)$ to record the number of times that the state sequence of this hypothesis has traversed each state. Here w_i^k is the number of times that this state sequence has traversed the state s_i up to the 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 Algorithm 3.1. If a hypothesis is removed at time $t=k+1$, then the vector of this hypothesis is abandoned. Otherwise assuming that the new hypothesis $\Omega_v^{k+1}=(\Omega_i^k,s_j)$, we can update the vector W^{k+1} with $W^{k+1}=(w_1^k,\cdots,w_j^k+1,\cdots,w_N^k)$. At the termination time T, we have I^T valid hypotheses and I^T

vectors W_v^T ($1 \le v \le I^T$). Now if we analyze all of these I^T vectors, at least we can conclude whether all these hypotheses have traversed certain states or not. For example, as shown in Fig. 3, all hypothesis must have visited the state s_2. In fact by comparing these vectors, we know the minimal and maximal number of visits for each state. In some detection problems, some specific states represent "critical" events and an alert should be generated if we are sure that these states were traversed based on all possible hypotheses. The total visit number or frequency of each state could also be used in some detection problems.

This is just one example of the hidden information that we can extract from these vectors. For different detection problems, other useful information could also 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 $U^k = \{u_{ij}^k\}_{N\times N}$ to record the state transition history for each hypothesis. Here u_{ij}^k is the number of times that the state s_i transferred to the state s_j up to time $t = k$. U^k can be recursively computed in the same way as W^k. W^k is a vector and U^k is a matrix. They are both inherent properties of each hypothesis so that we name W^k and U^k as *first-order* and *second-order* properties of hypotheses, respectively. Similarly, by comparing I^T matrices U_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 the second-order properties U_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 also develop higher order properties to analyze hypotheses if necessary.

The 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 assume that all hypotheses of the weak model λ_2 do not visit any state that represents that specific event. Based on other information source, if we know that the event happens, 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 lead to identify the unique characters of a dynamic process.

5. SIZE OF HYPOTHESIS SET

As discussed earlier, for each hypothesis, these multi-order properties can be recursively computed with polynomial time. However, given an observation sequence O^t, a weak model may have a large number of hypotheses resulting from Algorithms 3.1 or 3.2 and the size of hypothesis set may grow exponentially along the time. 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. Straightforwardly the computing complexity of hypothesis statistical analysis is roughly proportional to the size of the hypothesis set.

As mentioned in Section 3, we denote I^k as the total number of hypotheses at time k. Further we denote I_i^k ($1 \le i \le N$) as the number of hypotheses ending with the last state $s^k = s_i$ and we have $I^k = \sum_{i=1}^{N} I_i^k$. For convenience, we 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 can use the following recursive algorithm based on dynamic programming to calculate the size of hypothesis set.

Algorithm 5.1:

1) Initialization: at time $t = 1$, with the observation o^1, we have
$$I_i^1 = B(s_i, o^1), 1 \le i \le N; \quad (7)$$
2) Induction: at time $t = k+1$, with the new observation o^{k+1}, we have
$$I_j^{k+1} = \left[\sum_{i=1}^{N} (I_i^k \cdot a_{ij})\right] b_j(o^{k+1}), 1 \le j \le N, 1 \le k \le T-1;$$
$$I^k = \sum_{j=1}^{N} I_j^k; \quad (8)$$
3) Termination: at time $t = T$, we have I^T valid hypotheses,
$$I^T = \sum_{j=1}^{N} I_j^T. \quad (9)$$

The size of hypothesis set can be recursively calculated with Algorithm 5.1 because the elements in A and B are either one or zero. This algorithm is an efficient solution to the Problem 1 proposed in Section 3 if we do not need to know the hidden state sequence. 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 $R^k = (r_1^k, r_2^k, \cdots, r_N^k)$ and r_i^k represents the set size of q_i^k described in Section 3. Based on these vectors R^k, a similar backward procedure can be developed based on dynamic programming to compute the size of hypothesis set even 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 rather than the whole state sequence, i.e., at time $t = k(k > L)$, all hypotheses ignore the differences of their state sequence segments that are earlier than $t = k - L + 1$. For Algorithm 3.2, it is very efficient to update the latest L-step hypothesis set by stopping the backward procedure at $t = k - L + 1$ instead of $t = 1$. For Algorithm 3.1, there are two approaches for updating the hypothesis set:

1) For time step $t = k(k > L)$, after step 2 of Algorithm 3.1, we can run a procedure to prune the hypotheses. The prune procedure is to cut the early part $(t \le k - L)$ of the state sequence off and keep the state sequence starting from $t = k - L + 1$ to $t = k$ for each hypothesis. Then we can 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:
$$\Omega_v^{k-L+1} = \left\{ s_i \mid I_i^{k-L+1} > 0 \right\}, 1 \le i \le N,\ 1 \le v \le I^{k-L+1}. \quad (10)$$
Then run step 2 of 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 much more efficient than the second one. Otherwise the second procedure is more efficient. In some cases, we do not 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 procedure can be applied to prune hypothesis set after the size of hypothesis set is bigger than a selected threshold. Algorithm 3.2 does not have these problems so that it is better to use Algorithm 3.2 when the time window is shifting.

6. STRUCTURE OF EMISSION MATRIX

Given an observation sequence O^t, the size of hypothesis set is determined by the structure of A and B. In general the size of hypothesis set will be small if the matrix A and B are sparse. Essentially the state transition matrix A is determined by the inherent 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 can be added to observe the states of the dynamic process. In this section we analyze how the size of hypothesis set can be reduced by adding sensors and tuning the structure of the emission matrix.

Define a vector $\Phi^k = (I_1^k, I_2^k, \cdots, I_N^k)^T$. Equation (8) can be rewritten with the following format:

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

where $D(o^k)$ is the diagonal matrix in Equation (11) and $D(o^k) \in \{D(o_1), D(o_2), \cdots, D(o_M)\}$. For convenience, we denote $D^k = D(o^k)$. The form of Equation (11) is usually viewed as a linear, time-variant and zero input system in control theory literature. According to the norm-1 definition, we have $\left\| \Phi^k \right\|_1 = \sum_{i=1}^N I_i^k = I^k$. An interesting question is: given the state transition matrix A of a dynamic process, can we choose a structure of the emission matrix B to make $\left\| \Phi^k \right\|_1$ 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, in fact we can always find an emission matrix B so that the size of hypothesis set will not grow exponentially under any sequence of observations. For example, each column of B only has one element with value one, i.e., $\sum_{i=1}^N b_i(o_j) = 1$ for any $1 \le j \le M$. With this emission matrix, the observation sequence can be mapped to the state sequence deterministically and we have only one hypothesis - the real state sequence. In practice we may need a large number of sensors to achieve that requirement and we also may not be able to observe some states directly. Here a challenging question is: given a specific structure of matrix A, what is the necessity condition of B to make the size of hypothesis set not grow exponentially?

Consider the worst case: each element of A is one, i.e., any state can transfer to another state and the structure of A does not place any constraints on state transitions. Multiplying a vector $[1,1,\cdots,1]_N$ to each side of Equation (11), we can have:

$$I^k = \sum_{i=1}^N I_i^k = \left[b_1(o^k), \cdots, b_N(o^k) \right] \left[I^{k-1}, \cdots, I^{k-1} \right]_N^T = \sum_{i=1}^N b_i(o^k) I^{k-1} \quad (12)$$

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

$$(b_{\min})^{k-1} I_1 \le I^k \le (b_{\max})^{k-1} I_1 \quad (13)$$

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

Here we consider another simple case: assume that each column of B is same, i.e., the deployed sensors are totally blind in distinguishing the states of a dynamic process. In this case, $D(o_i)$ is same for $1 \le i \le N$ and denote $D = D(o_i)$. Equation (11) can be viewed as a time-invariant linear system and rewritten with $\Phi^k = \left(DA^T\right)^{k-1}\Phi^1$. With this equation, it is well known that the stability of Φ^k is determined by the eigenvalues of the matrix DA^T. If DA^T has N distinct real eigenvalues $\lambda_i (1 \le i \le N)$ and $\max_i |\lambda_i| < 1$, Φ^k asymptotically decreases. If DA^T has other forms of eigenvalues, see the specific stability conditions in [8].

In most cases, the emission matrix B has different columns so as to distinguish states. We can have the following equations by using Equation (11) recursively:

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

$$\left\|\Phi^k\right\| = \left\|\prod_{t=2}^{k}(D^t A^T)\Phi^1\right\| \le \left\|\prod_{t=2}^{k} D^t A^T\right\|\left\|\Phi_1\right\| \quad (15)$$

For any possible sequence $(D_k, D_{k-1}, \cdots, D_2)$, if these exits a sufficient large k such that $\left\|\prod_{t=2}^{k} D^t A^T\right\| < 1$, Equation (15) is a contraction mapping and $\left\|\Phi^k\right\|$ ($\left\|\Phi^k\right\|_1 = I^k$) decreases. Note that $D^k = D(o^k)$ has to be one of the M matrices $\{D(o_1), D(o_2), \cdots, D(o_M)\}$ 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 [1][2][14]. However, those conditions are very strong and it is not clear what the structure of B should be to satisfy those conditions. Meantime, denote a finite set of matrices $\Sigma = \{D(o_1)A, D(o_2)A, \cdots, D(o_M)A\}$. We have proved in [4] that the hypothesis growth is polynomial if the joint spectral radius [3] of Σ is less than or equal to one. Conversely the hypothesis growth is exponential if the joint spectral radius of Σ is larger than one. However, since the joint spectral radius only converges to a value at infinite time, we still do not know the necessary structure of the emission matrix to meet this condition. We will analyze this problem in our future work.

7. APPLICATIONS

As mentioned earlier, weak models can be used to describe a wide class of dynamic processes. We have applied weak models to correlate distributed events for network security [9]. Computer networks produce large amount of event-based data that can be collected for network security analysis. Alerts from firewalls and Intrusion Detection Systems (IDS), software log files, system audit events and network traffic statistics are typical examples of such data. Network events are instantaneous occurrences of certain types of network activity at a point in time and location. If we regard computer networks as dynamic systems, network events are the observables of their dynamic state transitions. Given the distributed nature of networks, evidence of malicious attacks is often scattered across distributed systems and observation time. A critical challenge is how to correlate these events across observation time and space to detect various attacks. We use a codebook correlation matrix [17] to correlate events from distributed sensors spatially and then employ a weak model to correlate distributed events temporally [9]. Without loss of generality, in this section we use a simple example to illustrate how weak models can be applied in process detection problems.

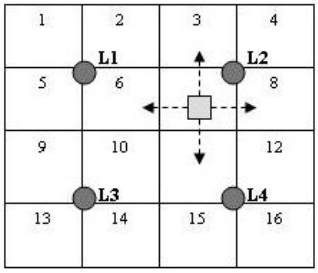


Fig. 4. A grid network under surveillance.

Fig. 4 shows a 4×4 grid network that is under surveillance. The square represents a moving target and the four circles represent four surveillance sensors deployed in this area. At each time step, the target moves to one of its neighbor cells, i.e., the target can randomly move left, right, up and down. Each sensor can monitor the closest four cells in its neighborhood. For example, the sensor L1 can monitor the cells 1, 2, 5 and 6. The goal of the sensor network is to collaboratively locate the moving target in this grid network. Assume that each sensor can only report "1" or "0" to indicate whether the moving target is within its detection coverage, i.e., these four sensors are binary sensors.

Table 1: States, sensors and observations

States	Sensors				Observations
	L1	L2	L3	L4	
s1, s2, s5, s6	1	0	0	0	o1
s3, s4, s7, s8	0	1	0	0	o2
s9, s10, s13, s14	0	0	1	0	o3
s11,s12,s15,s16	0	0	0	1	o4

Define each cell as a state and totally we have 16 states named with $s_i\ (1 \le i \le 16)$, where i is the number of a cell in Fig. 4. Based on the kinematical constraints of the moving target, it's easy to obtain a 16×16 dimension state transition matrix A. Since at each time step the target can only move one step in one of the four directions, many state transitions are not possible and the matrix A is sparse. For example, the state s_1 can only transfer to the state s_2, s_5 and we have $a_{1j} = 0$ for any j not equal to 2 or 5. Now we can derive the emission matrix determined by the sensor layout in the Fig. 4. Table 1 illustrates the correlation relationship between states, sensors' outputs and observations. Based on all possible output combinations of the four sensors, here we only have four different observations $o_j\ (1 \le j \le 4)$. Note that due to the specific sensor layout in the Fig. 4, each time only one sensor outputs "1" and the other three sensors output "0". Based on the Table 1, it's easy to derive a 16×4 dimension emission matrix B.

Each sensor can not distinguish the four states in its detection coverage. If we correlate the outputs of the four distributed sensors spatially, it's difficult to locate the moving target precisely. Assume that we have received a three-step observation sequence: $O^3 = \{o_1, o_2, o_4\}$, we can use Algorithm 3.1 to correlate these observations temporally and conclude all hypotheses of state sequences underlying this observation sequence.

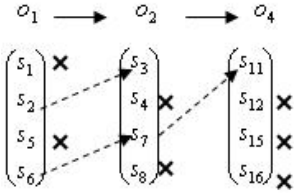


Fig. 5. Temporal observation correlation.

Fig. 5 illustrates the reasoning process of Algorithm 3.1. At the first step, with the observation o1, we have four state hypotheses. Then at the second step, with the observation o2, based on the kinematical constraints of the target, we know that the current state s^2 can only be s_3 or s_7 and only two hypotheses are possible (represented by the dotted lines). Furthermore, at the third step, with the observation o4, we can conclude that the current state s^3 must be s_{11} and there is only one hypothesis left. In fact at this point we also know where the target was at the earlier steps and this remaining hypothesis has to be the real process. Therefore the target location can be more precisely estimated by temporally correlating the observation sequence.

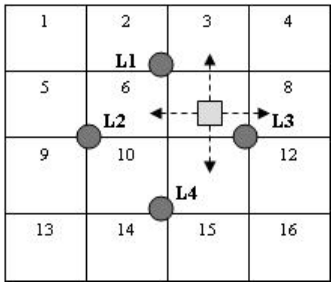


Fig. 6. The grid network with different sensor layout.

However, some observation sequences could make the size of hypothesis set grow exponentially. For example, given an observation sequence $O^t = \{o_1, o_1, \cdots, o_1\}$ (i.e. each observation is o1), with Algorithm 3.1, we can easily conclude that the hypothesis set grows exponentially along the time. In the same grid network, now we change the deployment of the sensors to the new layout as shown in Fig. 6. The only difference between scenarios illustrated in Fig. 4 and Fig. 6 is the location of sensors.

The state transition matrix A is not altered by the change of sensor locations. However, we need to re-derive the emission matrix determined by the new sensor layout. Table 2 shows the new relationship between states, sensors and observations. We have nine different observations with this new sensor layout. Based on the Table 2, we can easily obtain a new 16×9 dimension emission matrix B. If the target moves to the states s_6, s_7, s_{10} and s_{11}, it can be directly located by sensors without temporal reasoning. For any observation sequences that could generate in Fig. 6, with Algorithm 3.1, we can conclude that the size of hypothesis set will not be larger than two after the first step. Therefore, the new sensor layout can track the target much more precisely and the size of hypothesis set can be reduced dramatically by tuning the emission matrix. In general we should arrange sensor layouts to maximally distinguish states in process detection.

Table 2: The relationship with new sensor layout

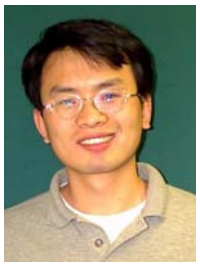
States	Sensors				Observations
	L1	L2	L3	L4	
s1, s4, s13, s16	0	0	0	0	o1
s2, s3	1	0	0	0	o2
s5, s9	0	1	0	0	o3
s6	1	1	0	0	o4
s8, s12	0	0	1	0	o5
s7	1	0	1	0	o6
s14, s15	0	0	0	1	o7
s10	0	1	0	1	o8
s11	0	0	1	1	o9

8. CONCLUSIONS

HMMs are widely used to model dynamic processes. For many process detection problems in defense and security applications, we do not have sufficient training data to precisely estimate the probabilities in HMMs. In this paper, we proposed a series of weak models to characterize dynamic processes. In weak models, we do not need the strong requirement for probability specification as in HMMs, which can dramatically reduce the difficulty and complexity of process modeling. Meantime, weak models are also not as sensitive to parameter estimation errors as HMMs and can be used to characterize non-stationary processes. We analyzed the mathematical 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 the size of hypothesis set can be reduced by tuning the structure of the emission matrix.

References

1. 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.
2. 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.
3. V.D. Blondel and J.N. Tsitsiklis, "The boundedness of all products of a pair of matrices is undecidable", *Systems and Control Letters*, 41:2, pp. 135-140, 2000.
4. V. Crespi, G. Cybenko and G. Jiang, "State sequence growth in nondeterministic finite automata", *Technical report*, Thayer School of Engineering, Dartmouth College, 2004.
5. G. Cybenko, V. Berk, V. Crespi, R. Gray and G. Jiang, "An overview of process query systems", in *Proc. of the SPIE Vol. 5403*, pp. 183-197, Orlando, FL, 2004.
6. S.R. Eddy, "Hidden Markov models", *Current Opinion in Structure Biology*, vol. 6, pp. 361-365, 1996.
7. Y. Ephraim and N. Merhav, "Hidden Markov processes", *IEEE Trans. on Information Theory*, vol. 48, no. 6, pp 1518-1569 , 2002.
8. O. Galor, *Introduction to Stability Analysis of Discrete Dynamical System*, Monograph in preparation, http://www.econ.brown.edu/fac/ Oded _ Galor.
9. G. Jiang and G. Cybenko, "Temporal and spatial distributed event correlation for network security", *Proc. of 2004 American Control Conference (ACC)*, pp. 996-1001, Boston, 2004.
10. B.H. Juang and L.R. Rabiner, "A probabilistic distance measure for hidden Markov models", *AT&T Tech J.*, vol. 64, no.2, pp. 391-408, 1985.
11. R.E. Kalman, "A new approach to linear filtering and prediction problems," *J. Basic Eng.*, vol. 82-D, pp.35-45, 1969.
12. V.H. Poor, *An Introduction to Signal Detection and Estimation*, Springer-Verlag, 1994.
13. L.R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition", *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257-286, February, 1989.
14. 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.
15. A. Wald, *Sequential Analysis*, John Wiley & Sons, 1947.
16. B.P.C. Warrender and S. Forrest, "Detecting intrusion using system calls: alternative data models", *1999 IEEE Symp. on Security and Privacy*, pp. 133-145, Oakland, CA, 1999.
17. S.A. Yemini, S. Kliger, E. Mozes, Y. Yemini and D. Ohsie, "High speed and robust event correlation", *IEEE Communications Magazine*, pp. 82-90, May, 1996.



Guofei Jiang received B.S. and Ph.D. degrees in electrical and computer engineering in 1993 and 1998 respectively from Beijing Institute of Technology, Beijing, China. From 1998 to 2000, he was a postdoctoral fellow at Thayer School of Engineering, Dartmouth College, NH.

He is a research staff member with NEC Laboratories America, Princeton, NJ 08540. From 2000 to 2004, he was a senior research scientist in the Institute for Security Technology Studies at Dartmouth College, working on several multi-million dollar government-funded projects. He has published over three dozen papers in distributed information systems, dependable and secure computing, machine learning, and system and information theory etc. He has served in the program committees of numerous conferences.