There exist many anomaly detection methods in the literature, which are based on Data Mining [3], Neural Network [4], Markov chains [5] and *etc*. However, we cannot find many methods in the literature that emphasize the large-scale web sites [6-7]. This paper uses the Hidden semi-Markov Model (HsMM) [8-9] to describe the user behaviors that may change with the time. Hidden Markov Model (HMM), which is widely applied in Speech Recognition, Character Recognition and DNA sequences clustering [8], is not widely used in the application of network security [10-12]. Some researches [13] have presented that network traffic has the property of second order self-similarity and long-range dependence. Yu *et al.* [14] prove that the HsMM is better than the HMM in describing the unstable distribution and can describe the second order self-similarity and long-range dependence of network traffic which may change with the time. Because of these advantages, the HsMM can be used to detect the anomalous user behaviors.

The rest of the paper is organized as follows. In Section 2, we introduce a parameter re-estimation algorithm of HsMM for multiple observation sequences. We then, in Section 3, propose two new on-line detection algorithms on user behaviors. In Section 4, we conduct an experiment using three sets of real traffic data to validate our detection algorithms. Finally, we conclude our work.

2. A PARAMETER RE-ESTIMATION ALGORITHM OF HSMM

The main difference between HsMM and HMM is that the state duration is not a constant or exponentially distributed. Ferguson [15] is the first to investigate estimation algorithms for the HsMM. However, Ferguson's algorithm is computationally too expensive to be of practical use in many applications. Yu *et al* [9] proposed a new forward-backward algorithm that reduces the computational complexity from $O((MD^2+M^2)T)$ to $O((MD+M^2)T)$, where M is the number of states; D the maximum possible interval between state transitions; and T the period of the observations used to estimate the model parameters. This new algorithm improves the computational efficiency of HsMM and promotes its applications. The algorithm is briefly reviewed as follows.

The parameters of HsMM are denoted as: $\boldsymbol{\lambda}=(\{a_{mn}\},\{\pi_m\},\{b_m(v_k)\},\{p_m(d)\})$, where a_{mn} is the state transition probabilities; π_m the initial state probabilities; $b_m(v_k)$ the observation element probabilities; $p_m(d)$ the probabilities of states' duration; o_t the observation sequence; $V=\{v_1,v_2,\ldots,v_k\}$ the set of observation elements; $S=\{1,2,\ldots,M\}$ the set of states; and $\{1,2,\ldots,D\}$ the set of state duration. Two forward-backward variables and other three variables denoted $\alpha_t(m,d)$, $\beta_t(m,d)$, $\zeta_t(m,n)$, $\eta_t(m,d)$ and $\gamma_t(m)$, are defined as follows:

$$\begin{aligned}\alpha_t(m,d) &\equiv \Pr[o_1^t,(q_t,\tau_t)=(s_m,d)] \\ &= \alpha_{t-1}(m,d+1)b_m(o_t)+\left(\sum_{n\neq m}\alpha_{t-1}(n,1)a_{nm}\right)b_m(o_t)p_m(d)],\end{aligned} \tag{1}$$

$$\beta_t(m,d) \equiv \Pr[o_{t+1}^T|(q_t,\tau_t)=(s_m,d)]$$

$$\hat{\pi}_i = \sum_{l=1}^{L}[\frac{1}{P[o^{(l)}{}_1^{T_l} \mid \lambda]} \cdot \gamma_1^{(l)}(i)] \Bigg/ \sum_{l=1}^{L}[\frac{1}{P[o^{(l)}{}_1^{T_l} \mid \lambda]} \cdot \sum_i \gamma_1^{(l)}(i)], \tag{12}$$

$$\hat{a}_{ij} = \sum_{l=1}^{L}[\frac{1}{P[o^{(l)}{}_1^{T_l} \mid \lambda]} \cdot \sum_{t=1}^{T_l} \xi_t^{(l)}(i,j)] \Bigg/ \sum_{l=1}^{L}[\frac{1}{P[o^{(l)}{}_1^{T_l} \mid \lambda]} \cdot \sum_{t=1}^{T_l}\sum_j \xi_t^{(l)}(i,j)], \tag{13}$$

$$\hat{p}_m(d) = \sum_{l=1}^{L}[\frac{1}{P[o^{(l)}{}_1^{T_l} \mid \lambda]} \cdot \sum_{t=1}^{T_l} \eta_t^{(l)}(m,d)] \Bigg/ \sum_{l=1}^{L}[\frac{1}{P[o^{(l)}{}_1^{T_l} \mid \lambda]} \cdot \sum_{t=1}^{T_l}\sum_d \eta_t^{(l)}(m,d)], \tag{14}$$

$$\hat{b}_m(k) = \sum_{l=1}^{L}[\frac{1}{P[o^{(l)}{}_1^{T_l} \mid \lambda]} \cdot \sum_{t=1}^{T_l} \gamma_t^{(l)}(m)\delta(o_t^{(l)} - v_k)] \Bigg/ \sum_{l=1}^{L}[\frac{1}{P[o^{(l)}{}_1^{T_l} \mid \lambda]} \cdot \sum_{t=1}^{T_l}\sum_k \gamma_t^{(l)}(m)\delta(o_t^{(l)} - v_k)], \tag{15}$$

$$P[O \mid \lambda] = \prod_{l=1}^{L} P[o^{(l)} \mid \lambda] = \prod_{l=1}^{L} \sum_{m,d} \alpha_{T_l}^{(l)}(m,d), \tag{16}$$

Where $i, j, m \in S$, $d \in \{1,2,\ldots, D\}$ and $v_k \in V$. From the above re-estimation formulas, we have the HsMM with multiple observation sequences.

3. ANOMALY DETECTION ALGORITHM

Usually, we track a user access behavior to a server by observing the sequence of objects the user requests. “Think time” is another observable presenting the time spent by the user in browsing a special object. We use a two-dimensional random vector $\vec{x} = (objectID, interval)$ to describe the user behavior in this paper, where *objectID* is the index of an object in the server, *e.g.* an HTML page or an image file, and *interval* is the “think time” of the user in browsing the object. Therefore, the user access behavior can be considered as a process $\{\vec{x}_t : t = 1,2,...,T\}$, where $\vec{x}_t$ is the value of $\vec{x}$ taken at time *t*. For different users we can obtain different sample sequences of the stochastic processes from the access logs of the server.

We assume that the process $\{\vec{x}_t\}$ is controlled by an underlying semi-Markov process. For an HsMM whose parameters are given, each state (called hidden state) of the HsMM can be used to describe a sequence of operations of the user. Transitions of the hidden states can be considered as the user’s browsing behavior from one web page to another following links between the pages. Therefore, likelihood of normal users’ access sequences computed by the given HsMM can be used to construct a distribution of the likelihoods. In considering that most of the normal users take the similar actions to access the server, i.e., with the similar likelihoods of the access process, we can define the normal degree of the user behaviors according to this distribution. Using this normal degree, we can judge a user who is normal or anomalous. As an application of this method, we propose an approach for anomaly detection as shown in Figure 1.

where M is the number of the Markov states, and $h_m(i)$ are the coefficients that can be determined by Yule-Walker equations [16]. When a new observation element appears in the real data, we denote it as v'_{K+1}, the output probability of which can be considered less than that of v'_K. Therefore, we can use the linear-prediction to estimate the value of $b_m(v'_{K+1})$ by

$$b_m(v'_{K+1}) = -\sum_{i=1}^{p} h_m(i) b_m(v'_{K+1-i}) \qquad 1 \le m \le M, \tag{18}$$

After we obtain $b_m(v'_{K+1})$ for each state m, we need to make the output probability matrix to be normalized (i.e., the sum of each row is 1). When all these are finished, we can use this temporarily updated HsMM to estimate the likelihood of real data that have the new observation element. In this algorithm, each state has p coefficients, which determines the accuracy of the estimation. Because V' has been sorted by the frequencies, $b_m(v'_{K+1})$ of the new observation data is just related to the probabilities of several closest observation elements. Hence we only need to set p to be a small number that can increase the efficiency of computation with a good estimation.

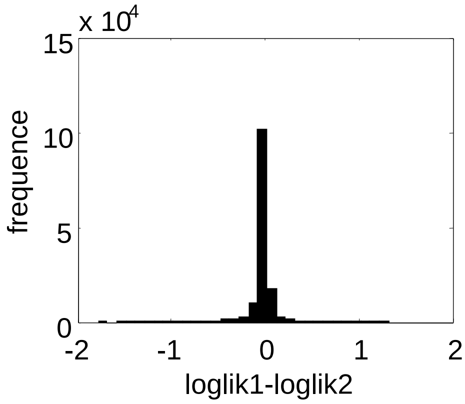
(2) On-line Algorithm of Updating Model Parameters

The above method can be used to estimate the probability of new observation elements which never appear in the training data set. However, the training data are finite and static, and user behaviors are changing with time. If the parameters of HsMM are also static without update, the model will become invalid gradually. For instance, if a new observation element v'_{K+1} appears with high frequency after some time (*i.e.* the majority of user behaviors are changing), and its accumulative total of frequency is more than that of v'_K, the system may still consider v'_{K+1} to be the element with minimum frequency. Apparently, the results (likelihoods) computed by this HsMM will deviate from the normal probability distribution built by the finite set of training data and become worse and worse. At last, we maybe get a wrong judgment for a normal user with new access behavior.

In order to solve this problem, the HsMM must be able to update its parameters with time. One solution could be like this: when a real sequence is input to the HsMM, the model uses both the original training data and this real sequence to re-estimate the parameters. Because of its large amount of computations, this solution is unpractical for on-line use. In this paper, we improve the re-estimation algorithm of HsMM for updating its parameters based on [17]. The main recursion process becomes as follows:

Let $\lambda^L = (\{a^L_{mn}\}, \{\pi^L_m\}, \{b^L_m(v_k)\}, \{p^L_m(d)\})$ be the parameters of HsMM with L training sequences and $\lambda^{(l)} = (\{a^{(l)}_{mn}\}, \{\pi^{(l)}_m\}, \{b^{(l)}_m(v_k)\}, \{p^{(l)}_m(d)\})$ be the parameters of HsMM which is trained by a single observation sequence l. Using the estimation algorithm of HsMM for multiple observation sequence described in section 2, we have:

$$a^{L+1}_{ij} = \frac{\sum_{l=1}^{L+1} \sum_{t=1}^{T_l} p[q_{t-1} = s_i, q_t = s_j \mid o^{(l)T_l}_1, \lambda]}{\sum_{l=1}^{L+1} \sum_{t=1}^{T_l} \sum_{j} p[q_{t-1} = s_i, q_t = s_j \mid o^{(l)T_l}_1, \lambda]} \equiv \frac{\sum_{l=1}^{L+1} trans(i,j,l)}{\sum_{l=1}^{L+1} states(i,l)}$$



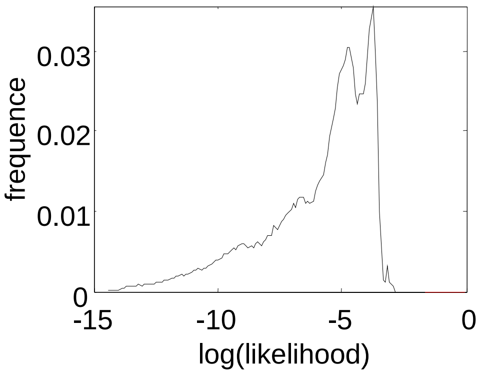


Figure 2 Distribution of *loglik1- loglik2*

Figure 3 Likelihood distribution of training data

We select three groups of data from [18] and denote them as *dataset1*, *dataset2* and *dataset3*. *dataset1* and *dataset2* include those users who may be individual users with low traffic volume; *dataset3* includes the users who may be proxy servers with the largest traffic volumes. Each user's sequence consists of two-dimensional random vector $\vec{x} = (objectID, interval)$.

In order to reduce the computational complexity, we map the two-dimensional vector into one-dimensional symbol.

(2) Checking the Linear-Prediction for Insufficient Training Data

We use *dataset1* to train the HsMM and obtain the parameters; and then we use this HsMM to compute the likelihood of *dataset2*. For those data which appear in *dataset2* but do not appear in *dataset1*, we use two methods to estimate their likelihoods respectively: (1) using the Linear-Prediction method introduced in the previous section and let p=2; (2) the HsMM uses both the input sequence and *dataset1* to estimate new parameters when the input sequence includes new observation elements, and then the new HsMM is used to compute the likelihood of this sequence. After the experiment, we obtain two groups of likelihoods of *dataset2*; we denote them as *loglik1* and *loglik2*. In order to analyze the difference of likelihood distribution between those two methods, we compute the distribution of (*loglik1-loglik2*), as shown in Figure 2. We can see that most of the values are close to the zero and the differences of the results obtained by these two methods are small. Therefore, we can use the Linear-Prediction method, whose computational complexity is more efficient than the other one in solving the Insufficient Training Data problem.

(3) Detection of Normal Degree

We select the *dataset1* to train the HsMM and obtain the OLD, as shown in Figure 3. The OLD is within [-14.3586, -2.9089], the mean of OLD is μ=-5.9653, and the variance is σ^2=4.7006.

In order to check the validity of our algorithm, we use the HsMM that is trained by *dataset1* to compute the likelihoods of both *dataset2* and *dataset3*, and compare the differences between them. In Figure 4, we can see that the differences of the likelihood distributions between *dataset1* and *dataset2* are small. This result shows that the users' behaviors in *dataset1* and *dataset2* are very similar. But in Figure 5, there exist significant

2. C. Manikopoulos and S. Papavassiliou, “Network Intrusion and Fault Detection: A Statistical Anomaly Approach,” IEEE Communications Magazine, October 2002, pp. 76-82.
3. G. Florez, S.A. Bridges, R.B. Vaughn, “An Improved Algorithm for Fuzzy Data Mining for Intrusion Detection,” Proceedings, 2002 Annual Meeting of the North American, Fuzzy Information Processing Society, NAFIPS 2002, pp.457-462.
4. S. Mukkamala, G. Janoski, and A. Sung, “Instrusion Detection Using Neural Networks and Support Vector machines,” Proceedings of the 2002 International Joint Conference on Neural Networks, IJCNN’ 02, Vol. 2, 2002, pp. 1702-1707.
5. D.-S. Xing and J.-Y. Shen, “A New Markov Model for Web Access Prediction,” Computing in Science and Engineering, Vol. 4, NO. 6, Nov./Dec. 2002,pp.34-39.
6. M. Arlitt and T. Jin, "A Workload Characterization Study of the 1998 World Cup Web Site," IEEE Network, May/Jun 2000, pp. 30-37.
7. Z. Liu, M. S. Squillante, C. Xia, S.-Z. Yu, L. Zhang, N. Malouch, "Analysis of Measurement Data from Sporting Event Web Sites," IEEE Globecom 2002, 17-21 Nov. 2002, Taipei, Taiwan, IPS-03-8.
8. L.R. Rabiner, “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition,” Proc. of IEEE, Vol. 77, No. 2, pp. 257-286, February 1989.
9. S.-Z. Yu, and H. Kobayashi, “An Efficient Forward-Backward Algorithm for an Explicit Duration Hidden Markov Model,” IEEE Signal Processing Letters, Vol. 10, No. 1, January 2003, pp. 11-14.
10. C. Warrender, S. Forrest, and B. Pearlmutter, “Detecting Intrusions Using System Calls: Alternative Data Models,” Proc. IEEE Symposium on Security and Privacy, pp. 133-145, 1999.
11. Y. Qiao, X. W. Xin, Y. Bin, and S. Ge, “Anomaly intrusion detection method based on HMM,” Electronics Letters, Vol. 38, No. 13, pp. 663-664, 20th June 2002.
12. Qing-Bo Yin, Li-Ran Shen, Ru-Bo Zhang, et al., “Intrusion Detection Based on Hidden Markov Model,” Proceedings of the Second International Conference on Machine Learning and Cybernetics, Xi’an, 2-5 November 2003.
13. A. Erramilli, M. Roughan, D. Veitch, and W. Willinger, “Self-Similarity Traffic and Network Dynamics,” Proceedings of the IEEE, vol. 90, no. 5, May 2002, pp. 800-819.
14. S.-Z. Yu, Z. Liu, M. S. Squillante, C. Xia, and L. Zhang, “A Hidden Semi-Markov Model for Web Workload Self-Similarity,” Proc. of The 21st IEEE International Performance, Computing, and Communications Conference (IPCCC 2002), pp. 65-72, April 3-5, 2002, Phoenix, AZ.
15. J. D. Ferguson, “Variable duration models for speech,” in Symp. Application of Hidden Markov Models to Text and Speech, Oct. 1980, pp.143-179.
16. Zhaoxiong Wu, Zhenxing Huang, Shunji Huang, “Digital Signal Process,”(Part 2), National Defence Public house, Dec. 1985, Edition 1.
17. Jinhui Xie, Liqing Gao, “The Relative measurement of HMM,” Automatization Transaction, 1993,19(5), 637-640.
18. http://ita.ee.lbl.gov/html/contrib/WorldCup.html