Optimizing Hidden Markov Model for Failure Prediction– Comparison of Gaine's optimization and Minimum message length Estimator

N.Muthumani
Department of Computer Applications
SNR Sons College
Coimbatore, India

Dr.Antony Selvadass Thanamani
Department of Computer Science
NGM College
Pollachi, India.

***Abstract*- Computer systems are prone to failures. Failures are caused by faults that occur in a system. As faults are unknown and cannot be measured, they produce error messages on their detection. The approach presented here is to create a Hidden Markov Model from the given data of error sequence and describes two techniques, Gaines algorithm and Minimum message length estimator to obtain a most appropriate Hidden Markov Model with optimized number of states. For a given sequence it is shown that both the two techniques ensure same optimal Hidden Markov Model with maximum probability.**

Keywords - Hidden Markov Models, Log files, Probabilistic finite state automata, Minimum message Length and emission probability.

I. INTRODUCTION

Nowadays we are highly dependent on the proper functioning of computer systems without failure. To design a fault tolerant system, the promising approach is failure prediction. Future Failures can be predicted by evaluating the current system state. So, it is essential to know the sources and frequency of errors, how faults manifest into errors, and which errors cause the failures. To determine preventive measures that allow downtime to be avoided error/event logs and system performance data can be used. The prediction of system failure requires a method to specify causal links between errors and failures. The indicators of failures can be generated a series of events marked with their occurrence times, logged in the system's log files The main principle of the approach presented here is to identify the frequency of errors and the relative probability that lead to failures. Hidden Markov Models (HMMs) have been successfully used in modeling. Examples include: speech recognition [1], genetic sequence analysis applications [2, 3 and 4], fault diagnosis, the detection of intrusion into computer systems and network traffic modeling [5, 6]. HMM is also used for failure prediction in a computer system [7]. This failure prediction is estimated using recent error events from the log files. As for failure prediction the output of hidden Markov models are probabilistic likelihoods. In this attempt an error sequence has been taken from the log files, a model is constructed and optimal path is identified using Minimum message length suggested by Georgeff and Wallace [8] for which the probability is found. The approach presented here consists of two steps. In the first step a model is built from recorded data using some training algorithm, which means that model parameters are adjusted such that some objective function is optimized. Specifically, training data consists of error-log files, which are used to identify whether a failure occurred or not. Having trained a model, the model is used to predict failures.

II. RELATED WORK

Liang et al. [9] predict failures of IBM's BlueGene/L from event logs containing reliability, availability and serviceability data. They use temporal and spatial compression. Temporal compression includes all events at a single location occurring with inter-event times lower than some threshold, and spatial compression includes all messages that refer to the same location within some time window. Berenji et al. [10] present a novel hybrid Model based and Data Clustering (MDC) architecture for fault monitoring and diagnosis, which is suitable for complex dynamic systems with continuous and discrete variables. Yang [11] presented a failure prediction method for preventive maintenance by state estimation using the Kalman filter. To improve preventive maintenance. They used a hybrid Petri-net modeling method coupled with fault-tree analysis and Kalman filtering to perform failure prediction and processing. Cheng et al. [12] proposed an application cluster service (APCS) scheme. The proposed APCS provides both a failover scheme and a state recovery scheme for failure management. [13] Turnbull et al. analyze hardware sensor data to predict failures in a high-end computer server. Hughes et al. [14] employ a rank sum hypothesis test to identify failure prone hard disks. Two improved

SMART algorithms are proposed. They use the SMART internal drive attribute measurements in present drives. The present warning-algorithm based on maximum error thresholds is replaced by distribution-free statistical hypothesis tests. Daidone et al. [15] have proposed to use a hidden Markov model approach. Taking advantage of the characteristics of the hidden Markov models formalism, widely used in pattern recognition, they proposed a formalization of the diagnosis process, addressing the complete chain constituted by monitored component, deviation detection and state diagnosis. This method is based on concurrent monitoring. So, this method could also be used for failure prediction: If a component is detected to be faulty, a failure is likely to occur. Weiss [16] introduces a failure prediction technique called "timeweaver" that is based on a genetic training algorithm. Timeweaver, a genetic-based machine learning system that solves the event prediction problem by identifying predictive temporal and sequential patterns within data. Leangsuksun et al. [17] describe that they have implemented predictive check pointing for a high-availability high performance Linux cluster. The dispersion frame technique (DFT) developed by Lin & Siewiorek [18] uses a set of heuristic rules on the time of occurrence of consecutive error events to identify looming permanent failures.

III. INTRODUCTION TO HIDDEN MARKOV MODEL

An HMM is mathematically equal to a stochastic finite automaton defined by a 5- tuple A = $(Q, \Sigma, \Delta, \pi, O)$ where $Q = \{ q_1, q_2, q_3, \dots\dots\dots, q_n \}$ is finite set of states , Σ is an alphabet of output symbols, $\Delta = \{ a_{ij} \mid 1 \leq i, j \leq n\}$ is a state transition probability distribution and $\pi = \{ \pi_i \mid 1 \leq i \leq n \}$ is an initial state distribution, O is the set $\{ e_j(x) \mid 1 \leq j \leq n \}$ of output symbol probabilities such that

$$\sum_{j=1}^{n} a_{ij}=1, \sum_{x\in\Sigma}^{n} e_j(x) \text{ and } \sum_{i=1} \pi_i = 1$$

HMMs are called "hidden" stems from the perspective that only the outputs can be observed from outside and the actual state q_i the stochastic process resides in is hidden from the observer. Three *basic problems* arise for which algorithms have been

1. Given a sequence of observations and a hidden Markov model, but having no clue about the states the process has passed to generate the sequence, what is the overall probability that the given sequence can be generated? This probability is called *sequence likelihood*.
2. Given a sequence and a model as above: What is the most probable sequence of states the process has traveled through while producing the given observation sequence?
3. Given a set of observation sequences: What are optimal HMM parameters A, B, and π such that the likelihood of the sequence set is maximal

IV. HIDDEN MARKOV MODEL FOR FAILURE PREDICTION

The objective is to assess the risk of failure for some time in the future. Here failures are predicted by analysis of error events that have occurred in the system. One assumption that is very common in failure prediction approaches in the notion that the frequency of error occurrence increases before a failure occurs.

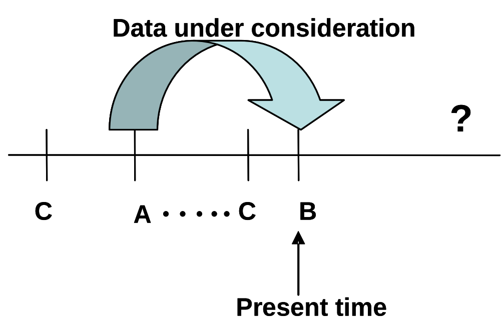

Fig. 1 Failure Prediction based on Error Events

In the above figure, Failure prediction is based on the occurrence of errors A, B, C. In order to perform the failure prediction, some data that have occurred shortly before present time are considered. In system trend analysis the keywords are defined as follows:

- Fault is an incorrect state of hardware or software.
- Symptom is observed out-of-norm parameter behavior.
- Error is manifestation of a fault observed by afault detector.

- Failure is one which occurs when the delivered service deviates from the specified service, i.e. failures are caused by errors.

HMMs have been shown to be successful pattern recognition tools in a large variety of recognition tasks ranging from speech recognition to intrusion detection in computer science. This being the first reason for the choice to use HMMs for failure prediction, second, referring to very basic distinction between faults, errors and failures. Faults are defined by unobserved. Once they manifest, they turn into errors, which are observable. This insight can be transferred analogously to HMMs. The states of an HMM are hidden i.e. unobservable, generating observation symbols. Hence, a close match exists between "hidden units", faults and the states of HMMs, and between their manifestations, which are errors and observation symbols, respectively. Given a sequence of observations a Hidden Markov model is successfully developed from a probabilistic finite state automata, then the overall probability of the given sequence can be found by sequence likelihood.

The HMM constructed here for a given data are restricted to have, from each state at most one transition with a given output symbol. Hence there exists exactly one path from initial state. The probability of the sequence is given by

$$P[I \mid A] = \pi_1 e_1(x_1)\ a_{12} e_2(x_2) \ldots\ldots e_{m-1}(x_{m-1}) a_{m-1,m} e_m(x_m),$$

$$\text{where } I = x_1 \ldots\ldots x_m$$

The error sequence AAACACBBBCCBBAACAAACB…… used here is a sample sequence assumed from the log files. Prediction of failures from a sequence of error events comprises two steps: first to fix the number of states in the automata and secondly to compute the probability that leads to failure.

This attempt is aimed to fit a model and to find the probability of the whole sequence in that model. First, let us construct the best automaton governing the above data well. There are a number of variations on HMM problems. When the number of states and its architecture are unknown, we find a HMM which models the data well. The simplest model has one state, the most complex model has a state for each and every symbol of the data but certainly neither extreme is justified [19].

The following are respectively 2, 3, 4 and 5 states automata that suit the output.

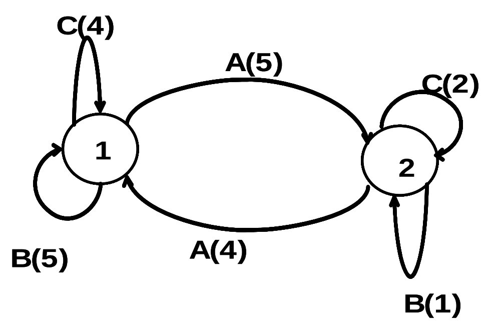

Fig. 2 Two States Automata

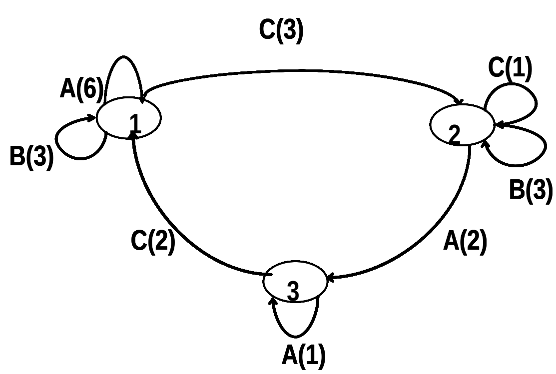

Fig. 3 Three States Automata

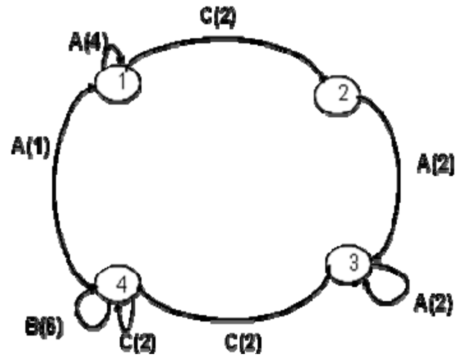

Fig. 4 Four States Automata

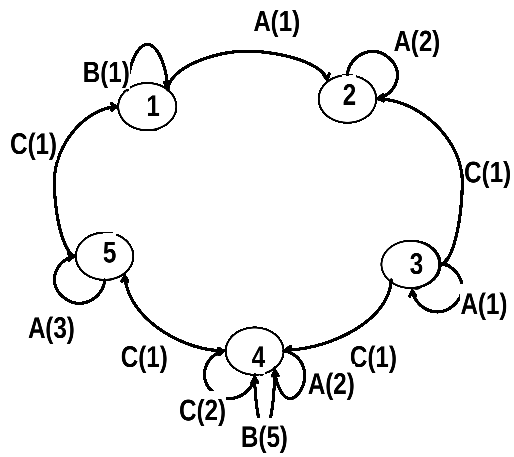

Fig. 5 Five States Automata

V. OPTIMIZATION OF HIDDEN MARKOV MODEL USING GAINES ALGORITHM

This optimization algorithm is drawn from Gaines [19] and best HMM is constructed from the given data. The number of states in the HMM is considered as the hypothesis and its reasonable measure with respect to the given data is computed. The Reasonable measure is the sum of the sum of all transitions of the log of the probability of the transition with a minus sign added with length of each state.Let v be the number of symbols in the data with probabilities $p_i, i = 1,2,3,...v$, and $\sum p_i = 1$. If a state is visited 't' times then the length of the state is given approximately by

$$\tfrac{1}{2}(v-1)\log(\tfrac{t}{12}+1)-\log(v-1)! - \tfrac{1}{2}\sum_i \log p_i$$

Here the logarithms are of base 2.

The probability p_i is estimated approximately as $(n_i + \tfrac{1}{2})/(t + \tfrac{v}{2})$, n_i is the number of times the i[th] symbol is produced from the particular state. The HMM optimized using Gaine's algorithm is called Gaine's HMM.

VI. MINIMUM MESSAGE LENGTH ESTIMATOR FOR HIDDEN MARKOV MODEL

An evaluation procedure is used to find the optimal HMM. The method is that of estimating the length of the message used to construct a best PFA model [8]. The same Minimum message length (MML) procedure is extended to identify best HMM.. The Minimum Message Length (MML) principle of Georgeff and Wallace has been used as an estimate here. In the context of PFSA, the MML is a sum of:

- the length of encoding a description of the PFSA;
- the length of encoding the data relative to the PFSA.

The formula used to compute the MML of a PFSA, originally derived by Raman and Patrick [20], is as follows:

$$\sum_{j=1}^{N}\left\{m_j + t_j + \log_2 \frac{t_j!(t_j-1)!}{(t_j-m_j)!(m_j-1)!\prod_{i=1}^{m_j} n_{ij}!}\right\} + M\log_2 V + M'\log_2 N + \log_2(N-1)!$$

Where:

- N is the number of states in the PFSA
- V is the number of tokens in the alphabet of arc labels
- t_j is the total of the frequencies on the arcs into the jth state
- m_j is the number of different arcs from the jth state
- m'_j is the number of different arcs on non-delimiter symbols from the jth state
- n_{ij} the frequency on the ith arc from the jth state _
- M is the total number of arcs in the PFSA

M' is the total number of arcs on non-delimiter symbols in the PFSA

VII. COMPARISON OF GAINES HMM AND MINIMUM MESSAGE LENGTH ESTIMATOR FOR A BEST HMM

Experimental results of the automata with states 2, 3, 4 and 5 constructed for given sequence are tabulated. Both procedures, Gaines Algorithm and Minimum message length estimator yields a consistent result in optimization. The reasonable measures for the above automata using Gaines Algorithm are given in Table 1.The reasonable measure starts with the value 36.5521 and increases gradually for the subsequent data, except for A_4. So A_4 is the optimum automaton as far as the given data is considered. Observations of Table 2 shows that the message length of A_4 is minimum after a gradual increase.

It can be calculated that the reasonable measure for each of the above automata respectively denoted by A_2, A_3, A_4 and A_5 are as given in the following table.

Automata	Message length
A_2	36.5521
A_3	36.7696
A_4	32.5843
A_5	37.4675

Table1 .Automata with RM

Automata	Message length
A_2	76.799
A_3	85.647
A_4	81.057
A_5	98.939

Table2 .Automata with MML

The HMM that best suits the given data is defined as A = (Q, Σ , Δ, π, O); where Q = { q_0 , q_1, q_2 , q_3 } , Σ = { A,B,C }, Δ = { a_{11} = 0.67, a_{12} = 0.33, a_{23} = 1, a_{33} = 0.5, a_{34} = 0.5, a_{41} = 0.11, a_{44} = 0.89 }, O = { $e_1(A)$ = 0.6, $e_1(B)$ = 0.07, $e_1(C)$ = 0.33, $e_2(A)$ = 0.72, $e_2(B)$ = 0.14, $e_2(C)$ = 0.14 , $e_3(A)$ = 0.45, $e_3(B)$ = 0.1, $e_3(C)$ = 0.45, $e_4(A)$ = 0.14, $e_4(B)$ = 0.62 , $e_4(C)$ = 0.24}, π = { π_1 = 1, π_2 = 0 , π_2 = 0, π_3 = 0 }.

The probability of I that is generated by A_2 is given by

$$\Pr(I/A_4) = \pi_1\, e_1(A)^4\, a_{11}^4\, e_1(C)^2\, c_{12}^2\, e_2(A)^2\, a_{23}^2\, e_3(A)^2\, a_{33}^2\, e_3(C)^2\, c_{34}^2\, e_4(B)^6\, b_{44}^6\, e_4(C)^2\, c_{44}^2\, e_4(A)\, a_{41}$$
$$= 9.078519\text{x } 10^{-14}$$

It can also be found that

$$\Pr(I/A_2) = 2.90407\text{x } 10^{-20}$$

$$\Pr(I/A_3) = 9.55897\text{x } 10^{-19}$$

$$\Pr(I/A_5) = 4.07118\text{x } 10^{-17}$$

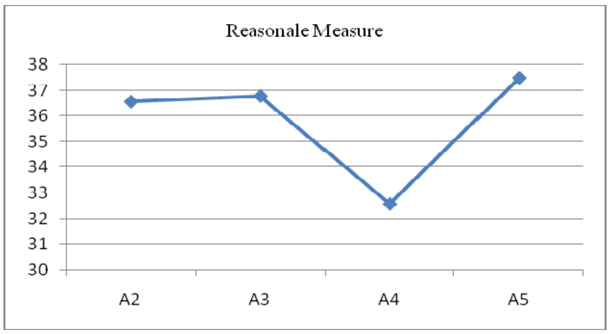

Fig 6. Reasonable Measures for the automata's

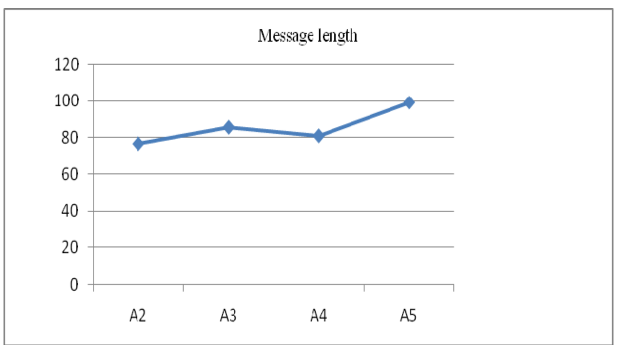

Fig 7. MML for the automata's

The probability of the sequence through A_4 is observed to be highest. Hence application of these two model search techniques shows effective and equivalent results in selecting an optimal model, say A_4 as the best optimal HMM.

VIII. CONCLUSIONS

In this paper, a new HMM based on Gaines algorithm and MML estimator for failure prediciton is presented using PFA for optimizing the number of states. It has been shown by comparison that the optimal model established for a given sequence by the above two techniques yields same selection and the best HMM provides highest probability. The future work suggested is to improve the probability of the sequence.

REFERENCES

[1] Rabiner L.R, A Tutorial on hidden Markov models and selected applications in speech recognition, *proc.IEEE.* vol. 77, pp. 257-286, 1989.

[2] Durbin, R., Eddy, S. R., Krogh, A., & Mitchison, G, Biological sequence analysis:probabilistic models of proteins and nucleic acids, *Cambridge University Press*, Cambridge, UK, 1998.

[3] L Steven Johnson, Sean R Eddy, Elon Portugaly, Hidden Markov model speed heuristic and iterative HMM search procedure, *BMC Bioinformatics*, 2010

[4] Yasubumi Sakakibara, Grammatical inference in bioinformatics, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 27, No.7, pp. 1051-1062, July 2005.

[5] Salfner, F, Predicting Failures with Hidden Markov Models, *In Proceedings of 5th European Dependable Computing Conference (EDCC-5),* 41–46. Budapest, Hungary,2005.Student forum volume.

[6] Salvo Rossi, P., Romano, G., Palmieri, F., & Iannello, G, A hidden Markov model for Internet channels, *In IEEE Proceedings of the 3rd International Symposium on Signal Processing and Information Technology (ISSPIT 2003),* 50–53, 2003.

[7] Warrender, C., Forrest, S., & Pearlmutter, B, Detecting intrusions using system calls: alternative data models, *In IEEE Proceedings of the 1999 Symposium on Security and Privacy,* 133–145. 1999.

[8] Georgeff M.P and Wallace C.S, A General selection criterion for inductive inference, *Advances in Artificial Intelligence,* North Holland, pp 219 - 228, 1984.

[9] Liang, Y., Zhang, Y., Sivasubramaniam, A., Jette, M., & Sahoo, R. BlueGene/L Failure Analysis and Prediction Models. In *IEEE Proceedings of the International Conference on dependable Systems and Networks (DSN 2006)*, 425–434. 2006

[10] Berenji, H., Ametha, J., & Vengerov, D. Inductive learning for fault diagnosis. In *IEEE Proceedings of 12th International Conference on Fuzzy Systems (FUZZ'03)*, volume 1.2003

[11] Yang, S. A condition-based failure-prediction and processing-scheme for preventive maintenance. *IEEE Transactions on Reliability*, volume 52(3): 373–383, 2003

[12] Cheng, F., Wu, S., Tsai, P., Chung, Y., & Yang, H. *Application* Cluster Service Scheme for Near-Zero-Downtime Services. In *IEEE Proceedings of the International Conference on Robotics and Automation*, 4062–4067. 2005

[13] Turnbull, D. & Alldrin, N. Failure Prediction in Hardware Systems. Technical report, University of California, San Diego,2003.

[14] Hughes, G., Murray, J., Kreutz-Delgado, K., & Elkan, C. Improved disk-drive failure warnings. *IEEE Transactions on Reliability*, volume 51(3): 350–357, 2002

[15] Daidone, A., Di Giandomenico, F., Bondavalli, A., & Chiaradonna, S. Hidden Markov Models as a Support for Diagnosis: Formalization of the Problem and Synthesis of the Solution. In *IEEE Proceedings of the 25th Symposium on Reliable Distributed Systems (SRDS 2006)*. Leeds, UK, 2006

[16] Weiss, G. Timeweaver: A Genetic Algorithm for Identifying Predictive Patterns in Sequences of Events. In *Proceedings of the Genetic and Evolutionary Computation Conference*, 718–725. Morgan Kaufmann, San Francisco, CA, 1999

[17] Leangsuksun, C., Liu, T., Rao, T., Scott, S., & Libby, R. A Failure Predictive and Policy- Based High Availability Strategy for Linux High Performance Computing Cluster. In *The 5th LCI International Conference on Linux Clusters: The HPC Revolution*, 18–20. 2004

[18] Lin, T.-T. Y. & Siewiorek, D. P. Error log analysis: statistical modeling and heuristic trend analysis. *IEEE Transactions on Reliability*, volume 39(4): 419–432, 1990

[19] Gaines B.R, Behaviour structure transformations under uncertainty, *Int J Man-Machines studies 8*, pp. 337-365, 1976.

[20] Raman AV, Patrick JD. Coding PFSA. Technical Report 1/97, Massey University Information Systems Department, Palmerston North, New Zealand, 1997