The representation of high order Markov process through equivalent first order process

Vitaliy Yakovyna, Oksana Nytrebych, Dmytro Fedasyuk

Software Department, Lviv Polytechnic National University, UKRAINE, Lviv, S. Bandery street 12,
E-mail: yakovyna@lp.edu.ua, ksenija.volynj@gmail.com, fedasyuk@lp.edu.ua

***Abstract* – The expanded transition probability matrix with size of $S \cdot \left(S^n\right) \times S$ is used to represent n-order Markov process, which consists of S components, through first order process. With growth of process order the matrix size increases rapidly and a lot of resources are needed to store it, although many of its elements are zero. In this paper we propose to split states of Markov processes to "fictitious" depending on the model order, which can significantly reduce the size of the transition probability matrix.**

Key words – software reliability, architecture software reliability models, higher order Markov process, transition probability matrix.

I. Introduction

The theory of Markov processes is widely used in problems of estimation and prediction of software reliability [1, 2]. It is proposed to use higher order Markov chains to take into account independent components execution in software, but the practical application of these chains in reliability models still remains an actual problem [3, 4].

An important applied aspect of reliability models based on Markov processes is the calculation of higher order transition probabilities between the system states. For first order processes such mathematical tools are well investigated and lead to the solution of Kolmogorov-Chapman differential equations in the case of continuous time [5, 6]. On the other hand any higher-order Markov process can be represented through the first order process [7, 8]. The algorithm should be formalized for representation of software implementation of models that use higher order chains. For example, in [7, 8] it is proposed to represent transition probability matrix of second order Markov process P^2, which contains S states, in the form:

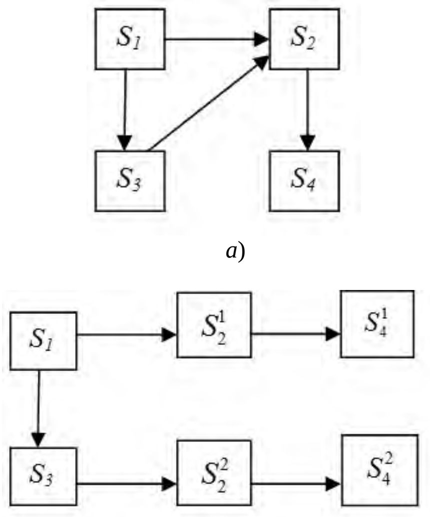
$$P^2 = \begin{pmatrix} p_{111} & p_{112} & \cdots & p_{11S} \\ p_{121} & p_{121} & \cdots & p_{12S} \\ \vdots & \vdots & \vdots & \vdots \\ p_{1S1} & p_{1S2} & \cdots & p_{1SS} \\ \vdots & \vdots & \vdots & \vdots \\ p_{SS1} & p_{SS2} & \cdots & p_{SSS} \end{pmatrix},$$

where p_{ijk} – transition probability from state S_i to state S_j, and then to state S_k. But due to the fact that there are no the second order routes between the states many elements of the matrix are filled by zeros, so this representation is not efficient in terms of program implementation as it requires lot of memory to store zero elements of the matrix.

II. The algorithm of representation of higher order Markov process through equivalent first order Markov process

Using the analogy with the known method of Erlang phases [9] it is possible to represent higher order Markov process as an equivalent first order process with additional fictitious states. In addition, each state of the initial graph (geometrical scheme shows the possible states of the system and the possible transitions between states in the system) is split into a number of fictitious states, which equals to the number of different paths to leading to this state. Thus, solution of the problem, in fact, is reduced to the usage the apparatus of graph theory.

As an illustration the proposed approach consider Fig. 1a. It shows the transition graph for a system of four components. Considering the second order Markov process, there are two routes leading to the state S_4: $S_1 \to S_2 \to S_4$ and $S_1 \to S_2 \to S_4$. According to the proposed approach, the state S_4 can be "splitted" into two fictitious states S_2^1 and S_2^2 and the system will then have the form as shown on Fig. 1b.



a)

b)

Fig. 1. The example of equivalent transformation of second-order Markov process to the first order process (*a* – the states graph of initial system, *b* – the states graph of system with fictitious states)

Let the M^k be a matrix, which elements m_{ij}^k denotes the mean number of possible routes of *k*-th order from state S_i to S_j.

Consider matrix M^1, elements of which denote the number of first-order transitions between appropriate states, shown in Fig 1a ($m_{12}^1 = 1$ is the element of matrix M^1, meaning that there is a single first order route from state S_1 to S_2):

$$M^1 = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

It is easy to show that matrix M^2, which contains number of routes from state S_i to S_j up to the second order, has following form:

$$M^2 = \begin{pmatrix} 0 & 2 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

where $m_{12}^2 = 2$ means that there is one second order route (second order Markov chain) $S_1 \to S_3 \to S_2$ and one first order route $S_1 \to S_2$ from state S_1 to S_2.

The matrix M^k has the following properties: the sum of all elements in the j-th matrix column is the number of k-order routes that enter state S_j (number of fictitious states S_j should be splitted to), and the sum of elements in the i-th matix row is the number of routes exiting from state S_i.

Therefore, it is possible to use a formula based on Floyd method for obtaining the number of k-order routes from state S_i to S_j:

$$m_{ij}^k = \sum_{l=1}^{S} m_{il}^1 (m_{lj}^{k-1} + \delta_{lj}), \quad (1)$$

where S is the number of all system states, δ_{ij} – Kronecker delta.

In this way we obtain an extended transition probability matrix, which is used as a basis for building Kolmogorov-Chapman differential equations system. In the case of variable order Markov process the different order value is used for each state, without changing the essence of the calculation process in formula (1).

Conclusion

The paper shows a new approach for representation of higher order probability matrix through the first order with a minimum number of zero elements. The advantage of the proposed approach is its clearness, applicability to Markov processes of any order (including variable), direct usage of the obtained extended matrix for building of Kolmogorov-Chapman differential equations system.

References

[1] K. Goševa-Popstojanova, S. Kishor, "Architecture-based approach to reliability assessment of software systems", *Performance Evaluation*, vol. 45, pp. 179–204, 2001.

[2] H. Pham, *System software reliability*. London: Springer-Verlag London Limited, 2006.

[3] T. Takagi, Z. Furukawa, T. Yamasaki, "Accurate Usage Model Construction Using High-Order Markov Chains", *Supplementary Proceedings of 17th International Symposium on Software Reliability Engineering*, pp. 1-2, 2006.

[4] Adrian E. Raftery, "A model for high-order Markov chains" *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 47, no 3, pp. 528–539, 1985.

[5] V. Tikhonov, V. Mironov, *Markovskie processy* [*Markov process*]. Moscow: Sovietskoe Radio Publ., 1977.

[6] Oliver C. Ibe, *Markov Processes for Stochastic Modeling*. Amsterdam: Elsevier Science & Technology Books, 2013.

[7] A. Berchtold, E. A. Raftery, "The mixture transition distribution model for high-order Markov chains and non-Gaussian time series", *Statistical Science*, vol. 17, no. 3, pp. 328–356, 2002.

[8] A. Shamshad, M. A. Bawadi, W.M.A. Wan Hussin, T.A. Majid, "First and second order Markov chain models for synthetic generation of wind speed time series", *S.A.M. Sanusi Energy*, vol. 30, no 5, pp. 693–708, April 2005.

[9] B. Yu. Volochiy, L. D. Ozirkovskyy, I. V. Kulyk, "Formalizacija pobudovy modelej dyskretno-neperervnyh stochastychnyh system z vykorystanniam metodu faz Erlanga [Formalization of construction of models of discrete-continuous stochastic systems using the method of Erlang phases]", *Vidbir ta obrobka informatsiji – Imformation Extraction and Processing*, no. 36 (112), pp. 39–47, 2012.