Alignment Graph Analysis of Embedded Discrete-Time Markov Chains

Dimitar Radev, Svetla Radeva

Abstract: A stochastic method for optimal graph alignment at analysis of embedded discrete-time Markov chain is presented. The method works by generating paths through a graph according to a Markov chain. Each path is assigned a score, and these scores are used to modify the transition probabilities of the Markov chain. This procedure converges to a fixed path through the graph, corresponding to the optimal (or near optimal) sequence alignment. Simulation and numerical results for the entrance probability vectors for tandem queue performance are shown.

Key words: Discrete Event System, Markov Chains, Queuing Theory, Communication System

INTRODUCTION

Sequence alignment is an approach for stochastic analysis of a dataset, presented as a string or a graph. There are various forms of sequence alignment. Alignments can be made between sequences of the same type or between sequences of different type. Pairwise alignment involves only two sequences, whereas multiple sequence alignment involves more than two sequences. Global alignment aligns whole sequences, whereas local alignment aligns only part of sequences [1].

Algorithms used for sequence alignment can be classified as deterministic, stochastic and heuristic [4]. Deterministic algorithms formulate sequence alignment as an optimisation problem and search deterministically for a global optimal alignment. Stochastic algorithms also formulate sequence alignment as an optimisation problem, but use stochastic optimisation techniques to search for a global optimum. Stochastic algorithms are often faster than deterministic ones, but have the disadvantage that they may return a sub-optimal alignment. Heuristic algorithms differ from stochastic algorithms in that they use a problem specific search method, rather than standard stochastic optimisation techniques. Heuristic algorithms also may return a sub-optimal alignment [6].

The classical approach to computing optimal alignment is via dynamic programming. The method presented here is a stochastic algorithm for pairwise global alignment. It uses an executing new technique for stochastic optimisation with the help of which quicker can be calculated the probability transition matrixes and can be received fast estimation of discrete-time Markov chains.

MARKOV CHAIN FRAMEWORK

Consider a Markov process (X_n : $n \ge 0$) where each X_n takes values in the state space S. For example, in a queuing network setting X_n may denote a vector of queue lengths just before the *n*-th change in the queue lengths, e.g., by an arrival and/or a departure from any queue. In such cases, where needed, residual service times and inter-arrival times may be appended as state descriptors to make (X_n : $n \ge 0$) a Markov process [3].

Let $(Y_n : n \ge 1)$ be an independently and identically distributed process where each $Y_n \in \Re^d$ (here \Re^d denotes the non-negative real numbers) and its components $(Y_n(i): i \le d)$ are independent of each other for each *n*. Further suppose that the evolution of $(X_n : n \ge 0)$ is governed by the process $(Y_n : n \ge 1)$ as follows:

➤ There exist non-negative integer valued random variables ($N_0(i)$: $i \le d$) that may depend only on X_0 such that X_1 is a deterministic function of X_0 and of random variables ($Y_m(i)$: $m = 1, ..., N_0(i)$) for all *i* for which $N_0(i) > 0$.

Similarly, for each *n* there exist non-negative integer valued random variables (N_n (*i*): $i \le d$) that may depend only on X_n such that X_{n+1} is a deterministic function of X_n and of random variables ($Y_m(i)$: $m = M_n(i) + 1, ..., M_{n+1}(i)$: $i \le d$) whenever $N_n(i) > 0$, where $M_n(i) = \sum_{i=1}^{n-1} N_j(i)$ for all *n* and *i*.

International Conference on Computer Systems and Technologies - CompSysTech'2004

For example, consider a *Gl/Gl/1* system observed at arrival and departure instants. In this setting let $X_n = (Q_n, RA_n, RS_n)$ where Q_n denotes the queue length just before the *n*-th change in its state, RA_n and RS_n denote, respectively the residual arrival and service times at this time. Sequence of random values $(Y_n(1) : n \ge 1)$ may denote the independently and identically distributed inter-arrival times and $(Y_n(2) : n \ge 1)$ may denote the sequence of independently and identically distributed service times. Here, the state X_{n+1} depends on X_n and on a random variable corresponding either to an inter-arrival time or a service time (or both, in case $Q_n = 0$ and $RA_n = 0$). In a queuing network, along with sequences of random variables denoting the inter-arrival and service times at each queue, $(Y_n : n \ge 1)$ may include other sequences of random variables that determine probabilistic routing of customers departing from the queues.

A simple interpretation and illustration of discrete-time Markov Chain framework can be given via two-dimensional stochastic node network considered on following example.

Example 1: *Steady state distribution of a tandem queue*. Consider the discrete time Markov process $X:=\{X_1(n), X_2(n)\}$, where $X_q(n)$, is the number of customers in queue q, q=1,2. The state space of the Markov chain is $S:=\{0,...,n_1\}\times\{0,...,n_2\}$. At the system can be defined following four events:

▶ *a* - queue 1 is not empty or full, e.g. $1 \le X_1 \le n_1$ -1

> *b* - queue 1 is empty, e.g. $X_1 = 0$

 \succ *c* - queue 1 is full, e.g. $X_1 = n_1$

> d - queue 2 is empty, e.g. $X_2 = 0$

The event *a* has occurred at one of three possible transactions: 1 - arrival in the first queue; 2 - service completion in the first queue; 3 - service completion in the second queue. The corresponding conditional probabilities for these transitions can be described as a_1 , a_2 , a_3 . However, when event *b* has occurred only two transitions are possible (1 and 3 of the list above). The corresponding probabilities are b_1 , b_3 . Analogically are defined c_1 and c_3 when queue 1 is full, and d_1 and d_2 when queue 2 is empty.

The one-step diagonal transition matrixes contain the one-step transition probabilities of the embedded discrete time Markov chains. As is shown in [5] can be defined three squared matrixes $A^{(q)}$ (q=1,2,3) of dimension (n_1+n_2+1) whose (i,j)-th element contains the probability that transition q occurs and the number of customers in the first queue changes from i to j. For each state are possible squared one-step matrixes in following three directions: *horizontal* to state (i+1,j) – matrix $A^{(1)}$, *diagonal* to state (i-1,j+1) – matrix $A^{(2)}$, and *vertical* to state (i,j-1) – matrix $A^{(3)}$. Diagonal elements of these squared matrixes are defined according to (1) and the rest of their elements are 0.

$$A^{(1)} = \begin{cases} a_{1} = \frac{\lambda}{\lambda + \mu_{1} + \mu_{2}} \\ b_{1} = \frac{\lambda}{\lambda + \mu_{2}} \\ d_{1} = \frac{\lambda}{\lambda + \mu_{1}} \end{cases} , \ A^{(2)} = \begin{cases} a_{2} = \frac{\mu_{1}}{\lambda + \mu_{1} + \mu_{2}} \\ c_{2} = \frac{\mu_{1}}{\mu_{1} + \mu_{2}} \\ d_{2} = \frac{\mu_{1}}{\lambda + \mu_{1}} \end{cases} , \ A^{(3)} = \begin{cases} a_{3} = \frac{\mu_{2}}{\lambda + \mu_{1} + \mu_{2}} \\ b_{3} = \frac{\mu_{2}}{\lambda + \mu_{2}} \\ c_{3} = \frac{\mu_{2}}{\mu_{1} + \mu_{1}} \end{cases}$$
(1)

On Figure 1 is shown stochastic node network for tandem queue with customers $n_{1=6}$ and $n_{2}=4$. With arrows are described all possible transitions at states of Markov chain. With small empty dots are depicted the states for which the second queue is empty and with big empty dots are depicted the states for which the second queue is full. A special occasion is the state $X_{6,4}$, shown with big black dot where the system is blocked because of overflowing of both queues.



Figure 1. Two-dimensional Markov chain of tandem queue

The steady state distribution analysis of these Markov chain is considering as probability of sequence transitions from one level to another, which can be realized as on rows or on diagonals of stochastic network. The up-crossing matrix Q(n) defines probability that starting from state *i* to state *j* at level *n*, level *n*+1 will be reached before level 0 and the entry state at level *n*+1 will be one with auxiliary coordinate *j*. The connection of up-crossing matrix and one-step transition matrixes is given by (2),

$$Q(n) = A^{(1)} + A^{(2)}Q(n) + A^{(3)}Q(n-1)Q(n) = \left(I - A^{(2)} - A^{(3)}Q(n-1)\right)^{-1}A^{(1)}$$
(2)

where *I* is the identity matrix.

SEQUENCE ALIGNMENT

An alignment of two sequences should reflect in some way the commonalities of the sequences if it is expected to be useful in applications. Some alignments are therefore better than others. This concept is formalized using a scoring function to assign a value to an alignment [2]. Let S_1 and S_2 be two sequences of length n_1 and n_2 , respectively, and let T_1 and T_2 be sequences of tokens obtained by inserting spaces into or at the ends of S_1 and S_2 such that T_1 and T_2 are of equal length *L*. Let $x = (T_1, T_2)$ represent an alignment of S_1 and S_2 . Conceptionally we can view (T_1, T_2) as a matrix with 2 rows and *L* columns where the first row contains the tokens of T_1 in order and the second row contains the tokens T_2 in order. Let V(x) be a scoring function on the space of all possible alignments. An optimal global sequence alignment is then an alignment x which solves (3).

$$\min V(\mathbf{x}) \tag{3}$$

Many of the scoring functions encountered in practice are of the following form. Each column *i* of the alignment is assigned a score $v(T_1(i), T_2(i))$, where $T_1(i)$ and $T_2(i)$ are the *i*-th tokens of T_1 and T_2 respectively and *v* is the so-called scoring matrix defined elements of summing (4).

$$V(x) = \sum_{i} v(T_1(i), T_2(i))$$
(4)

Alternatively, an alignment can be characterised as a path through a graph. Given two strings S_1 and S_2 of length n_1 and n_2 respectively, the edit graph for the string is the array of $(n_1 + 1)(n_2 + 1)$ nodes, each labelled with a distinct pair (i, j), $0 \le i \le n_1$, $0 \le j \le n_2$

and a set of *horizontal* (from (i,j) to (i+1, j)), *diagonal* (from (i, j), (i-1, j+1))and *vertical* (from (i, j) to (i, j-1)).

Moreover, any scoring function for alignment may be regarded as a scoring function for alignment paths, thus conferring a length to each path. The optimal alignment therefore corresponds to the shortest (or longest) alignment path. For scoring functions (4) one can associate weights to each edge in the edit graph: for *horizontal* edge ((i,j),(i+1, j)) the corresponding weight is v(S(i+1),-); for a *diagonal* edge ((i, j),(I-1, j-1)) the weight is v(S(i-1),S(i+1)) and *for vertical* edge ((i,j),(i, j-1)) the weight is; v(-, S(j-1)). Here, the " - " means space. The score of a given path is then the sum of the weights on its edges. An edit graph according to previous section is illustrated on following example.

Example 2: Sequence alignment for two queue networks. Define an alignment path through the edit graph from node (0,0) to node (11,4), that is, a sequence of edge-joined nodes as shown on Figure 2.



Figure 2. The graph path with sequence alignment

The alignment in Table 1 corresponds to the path from Figure 2, which has alignment vector with length L=31.

L	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
i	1	2	1	2	3	2	3	4	4	5	6	5	4	5	6	6
j	0	0	1	1	1	2	2	2	1	1	1	2	3	3	3	2
L	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
i	7	8	7	8	9	9	9	10	11	10	11	10	11	10	11	
j	2	2	3	3	3	2	1	1	1	2	2	3	3	4	4	

Table 1. An alignment vector from node (0,0) to node (11,4)

Furthermore, here is shown only of possible alignment vectors betweens considered nodes. The sequence alignment can be generated via random number generation and with the scoring function (4) to solve via combinatorial optimization the problem about shortest or longest path.

ENTRANCE PROBABILITY VECTORS WITH SEQUENCE ALIGMNENT

The basic implementation of discrete time Markov chain is calculating of the entrance probabilities, which for large enough models (e.g., models of networks with many queues) are receiving with enormous large matrixes.

The entrance probability vector $P_{m,k}$ at level *n* is defined as follows: its *i*-th element is the probability that the level *n* is reached before absorption (level 0) and that it is first entered at state *i*. Normalizing $P_{m,k}$ such that its elements sum up to 1 gives the entrance distribution of level *n*. On the other hand, summing the elements of $P_{m,k}$ gives the total probability of reaching level *n* before reaching 0. The starting state is given by its level *m* and a probability distribution over the states within that level. From this, the probability of reaching level k can be calculated as (5). $P_{m,k} = P(m).Q_1.Q_2...Q_{k-1}$

(5)

Let consider the optimal sequence alignment problem in terms of a shortest path through a graph. Specifically, in that context, we need to find an alignment path **x** through the edit graph for which the alignment score $V(\mathbf{x})$ is minimal. It is clear that this problem fits the combinatorial optimization. In particular, the state space *S* is given by the collection of all possible alignment paths. For illustration of this problem is considered the tandem queuing system of Example 1 with customers in firs queue $n_1 = 6$ and customers in second queue $n_2 = 4$.

Example 3: Total transition probability in tandem queue. Let the customers n_1 and n_2 in first and second queues have arrival rate λ =0,5 and departure service times μ_1 =0,4 and μ_2 =0,1. The state space of the Markov chain is *S*:={0,...,6}×{0,...,4} and Markov chain is an edit graph, starting at the bottom left-hand corner $X_{0,0}$ and finishing at the top right-hand corner $X_{6,4}$, e.g., describe the graph overflow probability. This problem is presenting with probabilities of level sequence transitions, where the diagonal levels are realized as a sum of customers n_1 and n_2 as is shown on Figure 3.





As is described in [5], the total conditional probability in each node can be presented with diagonal elements of horizontal, diagonal and vertical one-step transition matrixes (6).

$$P(X_{i,j}) = P([A_{i,j}^{(1)}]) + P([A_{i,j}^{(2)}]) + P([A_{i,j}^{(3)}])$$
(6)

With the help of indicator functions for horizontal $I_{\{i+1,,j\}}$, diagonal $I_{\{i-1\geq 0,j+1\}}$ and vertical $I_{\{i,j-1\geq 0\}}$ conditional probability easy can be calculated the general probability of the alignment path vector between two nodes, according to (7),

$$P(L_{i,j}) = \prod_{i=0}^{n_1} \prod_{j=0}^{n_2} I_{\{i+1,j\}} \Big[A_{i,j}^{(1)} \Big] + I_{\{i-1 \ge 0, j+1\}} I_{\{i,j \ne 0\}} \Big[A_{i,j}^{(2)} \Big] + I_{\{i,j-1 \ge 0\}} I_{\{i,j \ne 0\}} \Big[A_{i,j}^{(3)} \Big]$$
(7)

where indicator function $I_{\{i,j\neq 0\}}$ describes the events *a*, *b*, *c* and *d* defining the customers n_1 and n_2 . Let here calculate the probability for reaching node (6,4) from node (2,4) with alignment vector for shown on Figure 3 graph path. First is receiving the values for diagonal elements of squared one-step transition matrixes $A^{(1)}$, $A^{(2)}$ and $A^{(3)}$ as follows:

 $a_1=0,5; a_2=0,4; a_3=0,1; b_1=0,833; b_3=0,167; c_2=0,8; c_3=0,2; d_1=0,556 and d_2=0,444.$ Let the initial probability $P(X_{2,4})$ has value 2,5.10⁻² (can be determined with following alignment graph $P(X_{2,4}) = 1.d_1.d_2.a_1.a_1.a_2$). Than can be shown that the probability for reaching level 10 easy can be calculated as:

 $P_{4,10} = P(X_{2,4}).a_1.a_1.a_3.a_1.a_1.c_2.a_2.a_1.a_1.c_2.b_1 = 8,33.10^{-6}$

This example illustrates possibilities of suggested approach for fast and easy calculation of input probability distributions without calculation of large-scale matrixes using standard Markov chain theory. Naturally, for large enough models (e.g., models of networks with many queues) the size of calculation increases and become very large and the optimization problem have to be solved with operation research methods (Monte Carlo simulation, rare event simulation etc.).

CONCLUSIONS

In this work is suggested an approach for randomized sequence alignment, which is appropriate for combinatorial optimization of embedded discrete time Markov chains. The scoring function for graph alignment paths is minimized in correspondence with squared one-step transition matrixes in Markov chain. Numerical examples for fast calculation and easy estimation of entrance probability vectors between random nodes of stochastic network generated for tandem queue are presented.

The received results can found implementation for solving difficult combinatorial problems in telecommunication and computer networks.

The work is a part of the international research project PST.CLG.979333 with financial support of NATO scientific programs.

REFERENCES

[1] Bhulai, S. Markov Decision Processes: the control of high-dimensional systems, PhD thesis, Vrije University Amsterdam, 142 p., 2002.

[2] Garvels, M., D.P. Kroese. On the Entrance Distribution in RESTART Simulation. Proceedings of the RESIM'99 workshop, Twente, pp.65-88, 1999.

[3] Juneja, S., P. Shahabuddin. Efficient Simulation of Markov Chains with Small Transition Probabilities. Mamagament Science **47**, 4, pp.547-562, 2001.

[4] Keith, J., D.P. Kroese. SABRES: Sequence Alignment by Rare Event Simulation. Proceedings of the 2002, Winter Simulation Conference, San Diego, pp.320-327, 2002.

[5] Radev, D. Application of Cross Entropy for Quality of Service Estimation in Queuing Systems. Proc. of the Conference TELECOM'2003, Varna, pp.472-478, 2003

[6] Radev, D., S. Radeva. Kohonen Networks for Self-organizing Performance of Two Queues Markov Chains. Springer Lecture Notes in Artificial Intelligence, 2004.

ABOUT THE AUTHORS

Assoc. Prof. Dimitar Radev, PhD, Eng. Department of Communication Technique and Technologies, University of Rousse, Phone: +359 82 888 683, E-mail: dradev@abv.bg

Assoc. Prof. Svetla Radeva, PhD. Department of Computer Aided Engineering, University of Architecture Civil Engineering and Geodesy, Sofia, Phone: +359 2 9548 867, E-mail: svetla_fce@abv.bg