A Numerical Technique for the Identification of Discrete-State Continuous-Time Markov Models

L. S. Kuravsky, P. A. Marmalyuk, G. A. Yuryev and P. N. Dumin

Moscow State University of Psychology and Education, Computer Science Faculty, 29 Sretenka Street, 127051 Moscow, Russia

Copyright © 2014 L. S. Kuravsky et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract

A numerical technique for the identification of discrete-state continuous-time Markov models is presented. The technique is characterized by utilizing both initial approximations derived from observed data and estimates of minimized criterion sensitivity to small variations of identified parameters. A new feature of the proposed approach is improved running time of the applied combinatorial optimization algorithm achieved through replacing, at each iteration, of enumeration of various combinations of parameter values in the neighborhood of their current estimates by enumeration of values of only those parameters to which the minimized criterion is highly sensitive.

Keywords: Markov process, model identification, optimization procedure

1. Identification problem definition

The behavior of many engineering, diagnostic and other systems is represented by means of parametric mathematical models in the form of a set of ordinary differential equations. In particular, this form of representation was used for fatigue failure and computerized adaptive testing models based on discrete-state and continuous-time Markov random processes [2, 5, 6]. The important feature of Markov models practical application for the mentioned tasks is that model parameters are identified from observations represented by sets of values of examined functions at reference time points.

Markov models for describing dynamics of state-to-state transitions are represented by oriented graphs where nodes correspond to states and edges correspond to transitions for which the properties of Poisson event flows are met.
The number of events $N$ that fall in any time interval of length $\tau$, beginning at the time $t$, is distributed in these flows according to the Poisson law:

$$P_{t,\tau}(N) = \frac{a(t, \tau)^N}{N!} e^{-a(t, \tau)}$$

where $P_{t,\tau}(N)$ is the probability of occurrence of $N$ events during the considered interval, $a(t, \tau)$ is the average number of events falling in the interval of length $\tau$, beginning at the time point $t$. Only stationary flows are considered here where $a(t, \tau) = \eta \tau$, and $\eta$ is the constant transition rate of a stationary flow. The above assumptions for properties of event flows are usual for applied problems, as these flows (or flows which are similar in terms of properties) frequently occur in practice owing to the limit theorems for event flows.

It is assumed that

- For the specified continuous-time processes with $n$ discrete states, initial distributions of probabilities and observed frequencies of being in process states $\{F_{id}\}_{i=0,...,n}$ at time points $\{t_d\}_{d=0,...,D-1}$ are given, where $D$ is the number of time points at which frequencies $F_{id}$ are available; $0 \leq t_d \leq T$, where $T$ is the terminal time point.
- Transition flow rates between states are fully or partially unknown (free) parameters.

Dynamics of changing probabilities of being in process states is described by the set of Kolmogorov ordinary differential equations in matrix form:

$$\frac{d\mathbf{p}(t)}{dt} = \mathbf{M}(\lambda)\mathbf{p}$$

where $0 \leq t \leq T$, $\mathbf{p}(t) = (p_0(t), \ldots, p_n(t))^T$ are the probabilities of being in process states, $\lambda = (\lambda_0, \ldots, \lambda_m)^T$ is the ordered set of transition flow rates between states, $n > m + 1$, $\mathbf{M}$ is the matrix of transition flow rates between states of the order $n + 1$.

For the specified set of equations the problem of estimating a set of parameters $\lambda$ is formulated. Parameter values from this set are estimated by comparing observed and predicted histograms, which describe distributions of frequencies of being in model states, namely: the values are computed providing the best correspondence of observed and expected frequencies of falling into a certain system state at given time points. With that a set of transition flow rates $\lambda$ providing the least value of Pearson statistic is computed:

$$X^2(\lambda) = \sum_{d=0}^{D-1} \sum_{t=0}^{n-1} \frac{(p_t(\lambda, T_d)N - F_i)^2}{p_t(\lambda, T_d)N}$$
where \( N = \sum_{t=0}^{n-1} F_{id} \). This statistic is used as the model goodness-of-fit measure.

It has been proven [3] that under some common conditions values of the given statistic \( X^2 \) obtained after true solution substitution are asymptotically distributed as \( \chi^2 \) with \( n-m \) degrees of freedom, where \( m \) is the number of parameters to be identified. Moreover, estimated values of free parameters converge in probability to the target solution as sample size increases. This allows applying the Pearson statistic for verifying the hypothesis that obtained prediction fits observed data.

Thus, the problem assumes solving the inverse problem, where coefficients of differential equations are identified on the basis of partial solutions determined by observations. The above problem can be formally considered as the direct problem with a complicated computation of the \( X^2 \) criterion by solving the Cauchy problem for a set of ordinary differential equations, which is carried out by standard optimization procedures.

To solve this problem, in particular, local optimization algorithms, including gradient procedures, and stochastic algorithms are acceptable. However, the experience of their practical application [2, 6] demonstrates the unacceptably slow computations in case of large-scale problems (at larger values of the parameter \( m \)) as well as problems caused by probable non-uniqueness of a desired solution.

To solve these problems, the numerical optimization procedure for identifying a set of parameters \( \lambda \), which determines the matrix \( M \) and utilizes initial approximations computed from observed data, is proposed below. The basic version of this procedure belongs to the zero-order methods (without calculation of derivatives) while its modified version, which uses difference approximations of corresponding partial derivatives for evaluating the sensitivity of the minimized criterion to small variations of identified parameters, belongs to the first-order methods (with calculation of first-order derivatives). Proposed approach is further experimentally compared with a first order gradient method with adaptive steps [1], which is traditionally applied for solving multivariate nonlinear optimization problems.

2. Description of the technique

The proposed optimization procedure is based on the following computational algorithm, the principle of which for \( m = 2 \) is exemplified by Figure 1.

**Computational algorithm**

1. Using the available observation results, compute initial estimates of parameters \( \lambda_i \) \( (i = 1, ..., m) \) denoted as \( \lambda_i^0 \) \( (i = 1, ..., m) \) as sample means of number of transitions between pairs of the corresponding Markov process states per time unit (or, what is sometimes more convenient, quantities that are reciprocal to sample mean time duration between two adjacent transitions for the same pairs of states). Consider obtained estimates as the initial approximations to the identified parameters.
2. Assume $j = 1$, $k = 1$, $\beta_1 = 1 + \gamma_1$, where $\gamma_1$ is the algorithm parameter which can be set to 0.05 in many applications.

3. Compute the upper $\lambda_i^{j+}$ and the lower $\lambda_i^{j-}$ shift bound estimates from the formulas: $\lambda_i^{j+} = \beta_j \lambda_i^j$, $\lambda_i^{j-} = \beta_j^{-1} \lambda_i^j$ for each of current estimates of the identified parameters $\lambda_i^j$ ($i = 1, \ldots, m$) at $j$-iteration of the algorithm.

4. Having performed exhaustive enumeration of all possible variations of current estimates of a set of parameters $\lambda_j = (\lambda_0^j, \ldots, \lambda_m^j)^T$, in which each of its components $\lambda_i^j$ ($i = 1, \ldots, m$) takes only three possible values from the set \{${\lambda_i^{j-1}}, \lambda_i^{j-1}, \lambda_i^{j+}$\}, select one of the estimates $\lambda_{j,*} = (\lambda_0^{j,*}, \ldots, \lambda_m^{j,*})^T$ providing the minimal value of the $X^2$ criterion equal to $X_{2,\text{min}}^j$.

5. If the equality $\lambda_i^{j,*} = \lambda_i^{j-1}$ ($i = 1, \ldots, m$) is true for all the components of the chosen set $\lambda_{j,*}$, then assume $k = k + 1$, $\gamma_k = \gamma_{k-1}/2$.

6. Assume $j = j + 1$, $\beta_j = 1 + \gamma_k \cdot \lambda_j = \lambda_{j-1,*}$.

7. If $X_{j-1,\text{min}}^2 > X_2^*$, where $X_2^*$ is the algorithm parameter, then proceed to step 3, otherwise finish the computations.

---

**Figure 1.** Transition from iteration $j$ to iteration $j+1$ when performing steps 3-7 of the algorithm.
A numerical technique for identification

383

Computational complexity of an iteration of the algorithm under consideration is \(O(3^m)\), which results in unacceptably time-consuming computations at large values of \(m\) \((m > 6)\). For the computational complexity to become acceptable for practical purposes, the enumeration of variants of current estimates of a set of parameters \(\lambda_j = (\lambda^1_i, ..., \lambda^m_i)^T\) at step 4 of the above algorithm should be carried out not for all parameters \(\lambda^l_i\) \((i = 1, \ldots, m)\), but only for their subset consisting of \(l < m\) components, the variation of which strongly influence the \(X^2\) criterion. The modified computational algorithm takes the following form.

**Modified computational algorithm**

1. Using the available observation results, compute initial estimates of parameters \(\hat{\lambda}^i_l\) \((i = 1, \ldots, m)\) denoted as \(\lambda^0_i\) \((i = 1, \ldots, m)\) as sample means of number of transitions between pairs of the corresponding Markov process states per time unit. Consider obtained estimates as the initial approximations to the identified parameters.
2. Assume \(j = 1, k = 1\), \(\beta_1 = 1 + \gamma_1\), where \(\gamma_1\) is an algorithm parameter which can be set to 0.05 in many applications.
3. Evaluate the sensitivity of a \(X^2\) criterion in a small neighborhood of current estimates of identified parameters \(\lambda^j_i\) \((i = 1, \ldots, m)\) at \(j\)-iteration of the algorithm. For this purpose, use the difference approximations of partial derivatives absolute values \(\delta_i(\varepsilon) = \left| X^2(\lambda^0_l, ..., \lambda^j_i + \varepsilon, ..., \lambda^m_i) - X^2(\lambda^0_l, ..., \lambda^j_i, ..., \lambda^m_i) \right| / \varepsilon \), where \(i = 1, \ldots, m; \varepsilon\) is the small algorithm parameter.
4. Select \(l\) current estimates of identified parameters \(\lambda^j_{i_q}\), where \(i_q \in \{1, \ldots, m\}; q = 1, \ldots, l\); \(l\) is the algorithm parameter \((l < m)\) which have the largest values of sensitivity estimates \(\delta_q(\varepsilon)\).
5. Compute the upper \(\lambda^{j+}_{i_q}\) and the lower \(\lambda^{j-}_{i_q}\) shift bound estimates from the formulas \(\lambda^{j+}_{i_q} = \beta_j \lambda^j_{i_q}\) \(\lambda^{j-}_{i_q} = \beta_j^{-1} \lambda^j_{i_q}\) for each of chosen at step 4 current estimates of the identified parameters \(\lambda^j_{i_q}\) at \(j\)-iteration of the algorithm, where \(i_q \in \{1, \ldots, m\}; q = 1, \ldots, l\).
6. Having performed the exhaustive enumeration of all possible variations of current \(\lambda_j = (\lambda^1_j, ..., \lambda^m_j)^T\) estimates, in which each of its components \(\lambda^j_{i_q}\), where \(i_q \in \{1, \ldots, m\}; q = 1, \ldots, l\) takes only three possible values from the set \(\{\lambda^{j+1}_{i_q}, \lambda^{j-1}_{i_q}, \lambda^{j-1+}_{i_q}\}\) while the rest of components take the value of their current estimate at \(j\)-iteration of the algorithm, select one of the \(\lambda^j_{i,*} = (\lambda^1_{i,*}, ..., \lambda^m_{i,*})^T\) estimates providing the minimal value of the \(X^2\) criterion equal to \(X^2_{j, \min}\).
7. If the equality \(\lambda^j_{i,*} = \lambda^{j-1}_{i}\) \((i = 1, \ldots, m)\) is true for all the components of the chosen set \(\lambda^j_{i,*}\), then assume \(k = k + 1, \gamma_k = \gamma_{k-1}/2\).
8. Assume \( j = j + 1 \), \( \beta_j = 1 + \gamma_k \), \( \lambda_j = \lambda_{j-1} \ldots \).
9. If \( X^2_{j-1,\min} > X^2_\ast \), where \( X^2_\ast \) is the algorithm parameter, then proceed to step 3, otherwise finish the computations.

3. Computational experiments

Data generation
In order to assess computational efficiency of suggested algorithms a set of models was generated, which are structurally similar to models commonly used in practice. Essential structure of these models corresponds to the non-homogeneous birth-death process [8] such that state transition intensities are not equal for different state pairs in general. Controlled parameters of the model generation process were number of model states and number of independent parameters which values were selected from a predefined interval determining intensity values for all transitions in corresponding generated model.

The number of states varied in the range from 2 to 10 and number of independent parameters varied in the range from 1 to 18. There were 20 models generated for each possible combination of number of states and number of parameters, hence the total number of generated models was 1800. For all generated models the probability of being in the state \( S_0 \) at time point \( t_0 \) was equal to unity and zero for remaining states. Each generated model was numerically integrated by the 2nd order Euler method\(^1\) in the time interval \([0; 999]\) with the integration step equal to unity yielding a time series of probabilities of being in corresponding model states of length 1000.

Data obtained for each generated model had been saved into corresponding XML file. Each file includes: 1) model state names; 2) an adjacency matrix defining states linkage structure [11] and parameter-transition mapping; 3)

\(^1\) The following scheme of the second order modified Euler method, which was adopted for the equations of the birth-death process under consideration, was in use:

\[
p_k((z + 1)h) = p_k(zh) + \frac{1}{2}h[f_k[p_{k-1}(zh), p_k(zh), p_{k+1}(zh)] + f_k[p_{k-1}(zh) + hp'_{k-1}(zh), p_k(zh) + hp'_k(zh), p_{k+1}(zh) + hp'_{k+1}(zh)]],
\]

where \( k = 0, 1, \ldots, n \) is the number of a differential equation in the set (2) (if \( k = 0 \) then \( p_{k-1} \) in this expression is excluded, if \( k = n \) then \( p_{k+1} \) is excluded); \( h \) is numerical integration step; \( z \) is integration step number; the prime denotes the derivative with time; \( f_k […] \) is the expression that determines the derivative \( dp_k(t)/dt \) in the corresponding differential Kolmogorov equation of the set (2), i.e. "the right part" of this equation.
A numerical technique for identification

parameter values defining transition intensities values; 4) parameters of numerical algorithms; 5) time series $P(t) = (p_{S_0}(t), \ldots, p_{S_n}(t))$ of probabilities of being in model states at time points $\{t_d\}_{d=0}^{999}$. The archive “ModelData.zip” containing these XML files is available on the public repository at https://github.com/PMarmalyuk/Datafiles.

Computing system specifications

Software for computational experiment was implemented in G programming language using the National Instruments LabVIEW 2010 graphical programming environment [4]. Specifications of a computer, running on Microsoft Windows 7 Professional operating system (Service Pack 1), are the following:

- motherboard: ASRock Z77 Extreme4-M Socket LGA 1155 (1066 – 2800 MHz);
- quad-core CPU: Inter Core i5-3330 (3000 MHz);
- 2 RAM modules: Corsair XMS3 4 Gb DDR3 1333 MHz.

Identification procedure parameters

To avoid numerical problems associated with the digital representation of small probabilities, the statistic $Q = \sum_{d=0}^{D-1} \sum_{i=0}^{n} (p_i(t_d, \lambda) - \tilde{p}_i(t_d))^2$ was used as minimized criterion instead of $X^2$ statistics (3), where $\tilde{p}_i(t_d)$ is a probability of being in $i^{th}$ model state at the time point $t_d$. Optimization process stopped and corresponding model undergoing identification was considered as identified successfully when $Q$ reached the value satisfying the inequality $\frac{Q}{(n+1)D} \leq 0.001$, where $D$ is the number of time points forming model probability time series and $n + 1$ is the number of states. If the total number of iterations of optimization process exceeded 2000 or the total optimization time exceeded 200 seconds then corresponding model undergoing identification was considered as identified unsuccessfully.

Computational experiment results

Data sets with computational experiment results include the following information: 1) optimization method identificator; 2) number of states; 3) number of parameters; 4) iteration timestamps; 5) value of $Q$ reached at each iteration; 6) target value of $Q$; 7) model ID; 8) total time spent on model identification; 9) true and 10) numerically identified values of model parameters. These data were used to create a tidy dataset with overall results of the computational experiment. The
following variables were included into the dataset: a number of model states (States) and parameters (Params), model identification number (Trial), Euclidean distance from initial parameter estimates to their real values (Distance), number of iterations performed until one of the stopping criteria was satisfied (Iterations), resulting value of Q (ResFit), target value of Q (GoalFit), boolean flag (MissFitFlag) determining model identification successfully (0 – model identified successfully, 1 – model identified unsuccessfully) and categorical factor (Alg) determining numerical algorithm by which the results were obtained (0 – gradient-based method, 3 – enumeration of 3 significant parameters, 4 – enumeration of 4 significant parameters). Obtained data was analyzed with the aid of the R software environment for statistical computing [10]. The text file “results-all.txt” containing the dataset is also available on the public repository mentioned above.

The dependences of percentage of unsuccessfully identified models on number of parameters are presented in Figure 2 for different optimization methods. These results indicate that developed methods, with the exception of unmodified one for more than 5 parameters, provides significant acceleration of computations compared to the classical gradient method. The given dependencies show that the gradient method failed several times more frequently than suggested approaches (if number of parameters is greater than 8, solution cannot be determined in 34-55% of cases). This fact essentially limits application sphere of the gradient approach and, presumably, results from sensitivity of this way of solution with regards to selection of initial approximations.

![Figure 2](image-url)  
*Figure 2.* The relationships of percentage of unsuccessfully identified models on number of parameters for different optimization methods.
Number of iterations spent on model identification is considered sometimes as a comparison measure for the techniques in question. Figure 3 below contains panel box plots which present dependencies of the number of iterations on the number of model states (on the left) and on the number of model parameters (on the right).

![Figure 3](image)

**Figure 3.** Dependencies of the number of iterations on the number of model states (on the left) and on the number of model parameters (on the right).

Figure 3 shows that the gradient method is significantly inferior to the others in terms of the number of iterations spent on model identification. However, this fact demonstrates rather significant differences in the structure of computational algorithms (indeed, the modified algorithm includes an additional nested loop) than differences in computation time and efficiency.

Additional study of both the dependencies of the iteration average time on the number of model states (on the left) and on the number of model parameters (on the right), which are given in Figure 4, reveals that the iteration of the modified algorithm is executed far longer than the iteration of the gradient algorithm. Thus, comparison of the total number of iterations is not completely acceptable for the presented case.
Figure 4. Dependencies of the iteration average time on the number of model states (on the left) and on the number of model parameters (on the right).

To facilitate comparison of considered identification methods, dependencies of the average computation time on the number of model parameters are presented in Figure 5. The average computation time in case of the modified 3-parameter method is considered as 100% level. The obtained results show that the developed identification methods have advantages over the gradient approach when the number of identified parameters is not greater than 15. If this number is greater than 15, the gradient method is somewhat faster, however its convergence is not guaranteed in about a half of cases approximately (see Figure 2), and solution cannot be computed within a reasonable time.

Figure 5. The dependencies of the average computation time on the number of model parameters for considered optimization methods. The unit of measure (100% level) is the average computation time in case of the method based on the enumeration of 3 significant parameters.
It is not improbable that in some practical applications it will be reasonable to vary the $l$ parameter values at different computation iterations in order to avoid optimization “jam” at current estimates of identified variables. However, this scenario seems to be unlikely, because the parameters, variation of which is highly influential for the $X^2$ criterion, tend to decrease corresponding partial derivatives during optimization process, which leads to their elimination from a list which is corrected in a certain number of iterations.

In comparison with classical methods of unconstrained optimization including the pattern search method [1], the Nelder–Mead method [1], the cyclic coordinate descent method [4], the method of rotating coordinates [1] and the parallel tangents method [11], a new feature of the proposed approach is the improved running time of the applied combinatorial optimization algorithm achieved through replacing, at each iteration, of enumeration of various combinations of parameter values in the neighborhood of their current estimates by enumeration of values of only those parameters to which the minimized criterion is highly sensitive. Unlike the first two of the methods mentioned above the given approach presupposes dynamic changes of corrections applied to current estimates. As compared with the first-order optimization procedures, the method in question does not require highly accurate difference approximations of derivatives; it is sufficient to compute their rough estimates.

On the whole, presented results of the computational experiments demonstrate that, in the considered range of numbers of identified parameters, the developed modified identification method provides substantial speedup compared to the classical gradient method.

4. Main results

1. A numerical technique for the identification of discrete-state continuous-time Markov models has been developed. The given method is characterized by utilizing both initial approximations derived from the observed data and sensitivity estimates of the minimized criterion to small variations of identified parameters.

2. A new feature of the proposed approach is the improved running time of the applied combinatorial optimization algorithm achieved through replacing, at each iteration, of enumeration of various combinations of parameter values in the neighborhood of their current estimates by enumeration of values of only those parameters to which the minimized criterion is most sensitive.
3. Results of the computational experiments demonstrate that, in the considered range of numbers of identified parameters, the developed modified identification method provides substantial speedup compared to the classical gradient method.

References


A numerical technique for identification


Received: November 10, 2014; Published: January 6, 2015