A fast EM algorithm for fitting marked Markovian arrival processes with a new special structure

Gábor Horváth[1,2,3] and Hiroyuki Okamura[4]

[1] Budapest University of Technology and Economics
Department of Networked Systems and Services
[2] MTA-BME Information Systems Research Group
[3] Inter-University Center of Telecommunications and Informatics, Debrecen
1521 Budapest, Hungary
ghorvath@hit.bme.hu
[4] Department of Information Engineering
Graduate School of Engineering, Hiroshima University
1–4–1 Kagamiyama, Higashi-Hiroshima 739–8527, Japan
okamu@rel.hiroshima-u.ac.jp

Abstract. This paper presents an EM algorithm for fitting traces with Markovian arrival processes (MAPs). The proposed algorithm operates on a special subclass of MAPs. This special structure enables the efficient implementation of the EM algorithm; it is more orders of magnitudes faster than methods operating on the general MAP class while providing similar or better likelihood values. An other important feature of the algorithm is that it is able to fit multi-class traces with marked Markovian arrival processes as well. Several numerical examples demonstrate the efficiency of the procedure.

1   Introduction

Phase-type (PH) distributions and Markovian arrival processes (MAPs) play an important role in the performance and reliability analysis as they allow to describe a wide class of distributions and processes with Markovian techniques. The solutions of various queueing systems, failure models, etc. incorporating PH distributions and MAPs are typically numerically tractable.

However, the applicability of these models depends on how well the load of the system is represented, thus, how efficient the PH fitting and/or MAP fitting methods are when the empirical properties of the system are approximated.

Several fitting algorithms exist for PH distributions. As for correlated processes, the maturity of MAP fitting methods is a bit behind to the maturity of the PH fitting methods. There are several MAP fitting methods that aim to maximize the likelihood, all of them are based on the EM (expectation-maximization) algorithm. However, EM based MAP fitting algorithms have some distinct drawbacks that limit their practical usability. These algorithms suffer from slow convergence, high per-iteration computational effort and the final result is overly

dependent on the initial guess. As in case of PH fitting methods, it turned out that basing the fitting on a special MAP structure has a beneficial effect on both the convergence speed and the per-iteration computational effort. It is even typical that results obtained by fitting with general MAP structures are worse than those obtained by fitting with specialized structures [16].

The development of fitting procedures for the multi-type extension of MAPs, called MMAPs (marked Markovian arrival processes) is in initial stages, only a few solutions are available. In this paper we introduce a new special MMAP structure which enables a fast implementation of the EM algorithm. This special structure resembles to the ER-CHMM structure introduced for the single-class case in [16], but it is more general.

The rest of the paper is organized as follows. Section 2 introduces the notations and basic properties of marked Markovian arrival processes. We provide an overview on the available MAP and MMAP fitting methods in Section 3. The proposed MMAP structure and the corresponding EM algorithm is presented in Section 4. The numerical experiments are detailed in Section 5, finally, Section 6 concludes the paper.

2 Marked Markovian Arrival Processes

In a Markovian Arrival Process (MAP, [15]) there is a background process (also referred to as phase process) given by a continuous time Markov chain with generator matrix denoted by $\boldsymbol{D}$. Some of the transitions in this Markov chain are accompanied by an arrival event; the corresponding transition rates are the entries of matrix $\boldsymbol{D_1}$. The transition rates of the background process not accompanied by an arrival event are the entries of matrix $\boldsymbol{D_0}$. For the sum of these matrices we have $\boldsymbol{D} = \boldsymbol{D_0} + \boldsymbol{D_1}$ (we assume that $\boldsymbol{D}$ defines an irreducible Markov chain throughout this paper).

Marked Markovian arrival processes (MMAPs, [11]) are the multi-class extensions to MAPs, where the arrival events are tagged (marked) with the *class* of the arrivals. A MMAP distinguishing C different classes of arrival events is given by a set of matrices $\boldsymbol{D_c}, c = 0, \ldots, C$, where $\boldsymbol{D_0}$ describes the transition rates not accompanied by an arrival and $\boldsymbol{D_c}$ the ones accompanied by a type c arrival. The generator of the background Markov process is $\boldsymbol{D} = \sum_{c=0}^{C} \boldsymbol{D_c}$. Furthermore, we can introduce the embedded phase process at arrivals and obtain matrices $\boldsymbol{P^{(c)}} = (-\boldsymbol{D_0})^{-1}\boldsymbol{D_c}$ whose i, jth entry is the joint probability of the phase and the type of the next arrival given the phase at the previous arrival. The stationary phase probability vector at arrival instants, $\underline{\pi}$, is then the unique solution of $\underline{\pi}\boldsymbol{P} = \underline{\pi}, \underline{\pi}\mathbb{1} = 1$ with $\boldsymbol{P} = \sum_{c=1}^{C} \boldsymbol{P^{(c)}}$ ($\mathbb{1}$ denotes a column vector of ones of appropriate size).

Let us denote the random variable representing the inter-arrival time between the $i - 1$th and ith arrivals by $\mathcal{X}_i$, and the type of the ith arrival by $\mathcal{C}_i = \{1, \ldots, C\}$. The joint probability density function (pdf) that the consecutive inter-arrival times are $x_1, x_2, \ldots, x_K$ and the corresponding arrival classes are

$c_1, c_2, \ldots, c_K$ can be expressed by

$$f(x_1, c_1, x_2, c_2, \ldots, x_K, c_K) = \underline{\pi} e^{\boldsymbol{D_0} x_1} \boldsymbol{D_{c_1}} e^{\boldsymbol{D_0} x_2} \boldsymbol{D_{c_2}} \ldots e^{\boldsymbol{D_0} x_K} \boldsymbol{D_{c_K}} \mathbb{1}, \qquad (1)$$

which we will use several times in the sequel.

3 MAP and MMAP Fitting Methods

The MAP and MMAP fitting methods published in the literature can be classified to two categories. There are methods that are based on fitting statistical quantities of the trace (like moments, auto-correlations, etc.), while other methods aim to maximize the likelihood.

A purely moment matching based solution is described in [19], where a rational arrival process is constructed based on the marginal moments and the lag-1 joint moments of the trace. The resulting rational arrival process is transformed to a MAP in the second phase of the algorithm. This method has been generalized to MMAPs in [12]. A drawback of this approach is that it may happen that the result does not have a Markovian representation, or does not even define a valid stochastic process. An other popular framework falling into this category has been introduced in [13], where the MAP fitting is performed in two steps: the fitting of the marginal distribution and the fitting of the correlations. For the first step any PH fitting method can be applied, while for the second step several different solutions appeared. In [13] the target of fitting is the lag-k auto-correlation function, in [7] and [3] it is the lag-1 joint moments. The procedure in [6] is a MMAP fitting method based on marginal and joint moment fitting. The KPC procedure published in [8] follows a different approach, it achieves impressive results by combining small MAPs for fitting the moments, the auto-correlations and the bi-correlations of the trace.

Another family of MAP fitting methods aim to find a MAP or a MMAP that maximizes the likelihood of the measurement trace. The EM algorithm ([9]) is an iterative framework, which is a popular choice to implement likelihood maximization. Rydén [18] discussed an EM algorithm for a subclass of MAPs called MMPP (Markov modulated Poisson process), and it could naturally be extended to the general MAP parameter estimation. Buchholz [5] presented an improved EM algorithm in terms of computation speed for the general MAP class to analyze real trace data. The EM algorithms for the general MAP with multiple arrival types (or batches), i.e., the general MMAPs, were proposed in [4] and [14]. Okamura and Dohi [16] discussed the maximum likelihood estimation for the generalized structure.

4 Fitting Traces with MMAPs having a special structure

4.1 Motivation

A major drawback of EM-algorithm based MAP fitting methods is that they are slow if the number of measurements to fit is large. A large number of iterations

are required till convergence, and the per-iteration computational effort is also significant.

In case of EM-algorithm based phase-type (PH) distribution fitting methods the solution for this issue has been recognized for a long time. It turned out that fitting with sub-classes of PH distributions like hyper-exponential [10], hyper-Erlang [20], or acyclic PH distributions [17] is more efficient than fitting with the general PH class [2].

The computational effort of EM-algorithm based MAP fitting methods can be reduced in the same way, however identifying sub-classes of MAPs is far less trivial than it was in case of PH distributions. A possible sub-class of MAPs has been introduced in [13], where the proposed structure consists of a set of *component PH distributions* and a *transition probability matrix* that determines which component generates the next inter-arrival time given the current one, it is a kind of Markov-modulated PH distributions. (The procedure itself combines the moment matching and maximum likelihood estimation). Okamura and Dohi [16] provided a fast EM-based MAP fitting algorithm for the same structure, and found that it gives high likelihood values while the execution time is much lower compared with fitting by general MAPs. They found the case when the component PH distributions are Erlang distributions (called ER-CHMM) especially beneficial.

In this paper, our aim is to improve the ER-CHMM structure based EM-algorithm for fitting traces from several aspects:

– We generalize it to fit multi-type (marked) arrival processes as well.
– We introduce a structure that is more general than ER-CHMM, while the computational complexity remains relatively low.
– We present an improved method to optimize the discrete (shape) parameters of the structure.

4.2   The definition of the special structure used for fitting

In this section we define the special MMAP structure on which our fitting procedure is based on.

This special MMAP process is a generalization of the ER-CHMM structure. Similar to ER-CHMM, we have M branches with branch i consisting of r_i states connected in a row with the same transition rates λ_i. However, in our case these branches do not represent Erlang distributions, as not all states of the branch are traversed before generating an arrival. When a branch is selected to generate the next inter-arrival time, the initial state of the branch is determined by a probability vector.

Let us assign a two-dimensional identifier to the phases: phase (i, n) identifies state n in branch i.

The parameters characterizing this process are

– the rate and the shape parameters of the branches, denoted by λ_i and r_i, $i = 1, \ldots, M$, $\sum_{i=1}^{M} r_i = N$, respectively;

– probabilities $p_{i,(j,m)}^{(c)}$ with $c = 1, \ldots, C, \; i, j = 1, \ldots, M, \; m = 1, \ldots, r_j$. $p_{i,(j,m)}^{(c)}$ represents the probability that the next phase just after the arrival is (j, m) given that the previous arrival has been generated by branch i resulting in a type c arrival. Note that the ER-CHMM structure is obtained if $p_{i,(j,m)}^{(c)} = 0$ for $m > 1$.

According to the definition matrix $\boldsymbol{D_0}$ is given by

$$\boldsymbol{D_0} = \begin{bmatrix} -\lambda_1 & \lambda_1 & & & & & \\ & \ddots & \ddots & & & & \\ & & -\lambda_1 & & & & \\ & & & -\lambda_2 & \lambda_2 & & \\ & & & & \ddots & \ddots & \\ & & & & & -\lambda_2 & \\ & & & & & & \ddots \\ & & & & & & & -\lambda_M & \lambda_M \\ & & & & & & & & \ddots & \ddots \\ & & & & & & & & & -\lambda_M \end{bmatrix} \left. \begin{matrix} \\ \\ \\ \end{matrix} \right\} r_1 \left. \begin{matrix} \\ \\ \\ \end{matrix} \right\} r_2 \left. \begin{matrix} \\ \\ \\ \end{matrix} \right\} r_M \tag{2}$$

and matrices $\boldsymbol{D_c}, c = 1, \ldots, C$ are

$\boldsymbol{D_c} =$

$$\begin{bmatrix} 0 & \cdots & 0 & 0 & \cdots & 0 & & 0 & \cdots & 0 \\ \lambda_1 p_{1,(1,1)}^{(c)} & \cdots & \lambda_1 p_{1,(1,r_1)}^{(c)} & \lambda_1 p_{1,(2,1)}^{(c)} & \cdots & \lambda_1 p_{1,(2,r_2)}^{(c)} & \cdots & \lambda_1 p_{1,(M,1)}^{(c)} & \cdots & \lambda_1 p_{1,(M,r_M)}^{(c)} \\ 0 & \cdots & 0 & 0 & \cdots & 0 & & 0 & \cdots & 0 \\ \lambda_2 p_{2,(1,1)}^{(c)} & \cdots & \lambda_2 p_{2,(1,r_1)}^{(c)} & \lambda_2 p_{2,(2,1)}^{(c)} & \cdots & \lambda_2 p_{2,(2,r_2)}^{(c)} & \cdots & \lambda_2 p_{2,(M,1)}^{(c)} & \cdots & \lambda_2 p_{2,(M,r_M)}^{(c)} \\ \vdots & & & \vdots & & & \vdots & \vdots & & \\ 0 & \cdots & 0 & 0 & \cdots & 0 & & 0 & \cdots & 0 \\ \lambda_M p_{M,(1,1)}^{(c)} & \cdots & \lambda_M p_{M,(1,r_1)}^{(c)} & \lambda_M p_{M,(2,1)}^{(c)} & \cdots & \lambda_M p_{M,(2,r_2)}^{(c)} & \cdots & \lambda_M p_{M,(M,1)}^{(c)} & \cdots & \lambda_M p_{M,(M,r_M)}^{(c)} \end{bmatrix} \left. \begin{matrix} \\ \\ \end{matrix} \right\} r_1 \left. \begin{matrix} \\ \\ \end{matrix} \right\} r_2 \left. \begin{matrix} \\ \\ \end{matrix} \right\} r_M \tag{3}$$

By construction, the entries of matrix $\boldsymbol{P}^{(c)} = [u_{(i,n),(j,m)}^{(c)}, i, j = 1, \ldots, M, n = 1, \ldots, r_i, m = 1, \ldots, r_j]$ describing the transition probabilities of phases embedded just after arrival instants corresponding to type c arrivals are given by

$$u_{(i,n),(j,m)}^{(c)} = p_{i,(j,m)}^{(c)}. \tag{4}$$

The stationary distribution of the phases right after arrivals is the unique solution of the linear system

$$\underline{\pi} = \underline{\pi} \sum_{c=1}^{C} \boldsymbol{P}^{(c)}, \underline{\pi} \mathbb{1} = 1, \tag{5}$$

which can be partitioned according to the two-dimensional phase numbering as $\underline{\pi} = [\pi_{(i,n)}, i = 1, \ldots, M, n = 1, \ldots, r_i]$.

Given that the MMAP is in state (i, n) just after arrivals the density of the inter-arrival times is

$$f_{(i,n)}(x) = \frac{(\lambda_i x)^{r_i - n}}{(r_i - n)!} \lambda_i e^{-\lambda_i x}, \tag{6}$$

from which the marginal distribution of the arrival process is obtained by unconditioning yielding

$$f(x) = \sum_{i=1}^{M} \sum_{n=1}^{r_i} \pi_{(i,n)} f_{(i,n)}(x). \tag{7}$$

4.3 The EM Algorithm

Let us denote the trace data by $\mathcal{X} = \{x_1, c_1, x_2, c_2 \ldots, x_K, c_K\}$, where $x_k \in \mathbb{R}$ is the kth inter-arrival time and $c_k \in \mathbb{N}$ is the type of the kth arrival. In this section we assume that the shape parameters of the branches given by vector $\underline{r} = (r_1, \ldots, r_M)$ is fixed. Our goal is to maximize the likelihood for the trace data

$$L(\Theta(\underline{r}) | \mathcal{X}) = \underline{\pi} e^{\boldsymbol{D_0} x_1} \boldsymbol{D_{c_1}} e^{\boldsymbol{D_0} x_2} \boldsymbol{D_{c_2}} \ldots e^{\boldsymbol{D_0} x_K} \boldsymbol{D_{c_K}} \mathbb{1}, \tag{8}$$

where the parameters defining our MAP are $\Theta(\underline{r}) = \{\lambda_i, p_{i,(j,m)}\}$, from which matrices $\boldsymbol{D_c}, c = 0, \ldots, C$ are derived by (2) and (3).

Let the (i, n)th entry of row vector $\underline{a}[k] = \big(a_{(i,n)}[k], i = 1, \ldots, M, n = 1, \ldots, r_i\big)$ denote the likelihood of phase (i, n) after observing $x_1, c_1, x_2, c_2, \ldots, x_k, c_k$. This vector will be referred to as forward likelihood vector in the sequel and can be obtained recursively by

$$\underline{a}[0] = \underline{\pi}, \quad \underline{a}[k] = \underline{a}[k - 1] \cdot e^{\boldsymbol{D_0} x_k} \boldsymbol{D_{c_k}}. \tag{9}$$

Similarly, we can define backward likelihood vectors $\underline{b}[k] = \big(b_{(i,n)}[k]\big)$ as

$$\underline{b}[K] = \mathbb{1}, \quad \underline{b}[k] = e^{\boldsymbol{D_0} x_k} \boldsymbol{D_{c_k}} \underline{b}[k + 1]. \tag{10}$$

The likelihood is then given by

$$L(\Theta(\underline{r}) | \mathcal{X}) = \underline{a}[k] \cdot \underline{b}[k + 1], \tag{11}$$

for any $k = 0, \ldots, K - 1$.

Due to the special structure the forward and backward likelihood vectors can be expressed in a simpler way as

$$a_{(i,n)}[k] = \sum_{j=1}^{M} \sum_{m=1}^{r_j} a_{(j,m)}[k - 1] \cdot f_{(j,m)}(x_k) \cdot u_{(j,m),(i,n)}^{(c_k)}, \tag{12}$$

$$b_{(j,m)}[k] = \sum_{i=1}^{M} \sum_{n=1}^{r_i} f_{(j,m)}(x_k) \cdot u_{(j,m),(i,n)}^{(c_k)} \cdot b_{(i,n)}[k + 1]. \tag{13}$$

Note that as opposed to (9) and (10), (12) and (13) does not involve matrix exponential operations, which will be a significant gain in speed.

According to the EM approach we consider the data $\mathcal{X}$ to be incomplete, and assume that there is an unobserved data $\mathcal{Y}$ which, together with $\mathcal{X}$, forms the complete data. In our case the values of the unobserved data $\mathcal{Y} = \{y_1, z_1 \ldots, y_K, z_K\}$ inform us which branch generates the kth data item of $\mathcal{X}$ ($y_k \in \{1, \ldots, M\}$) and which was the initial state of branch y_k when generating the kth inter-arrival time ($z_k \in \{1, \ldots r_{y_k}\}$).

Given the unobserved data $\mathcal{Y}$, it is possible to obtain maximum likelihood estimates (MLE) for λ_i and p_{ij}. Let us start with λ_i. Since the inter-arrival times are independent given the unobserved data we can express the log-likelihood of parameters $\lambda_1, \ldots, \lambda_M$ as

$$\log L(\lambda_1, \ldots, \lambda_M | \mathcal{X}, \mathcal{Y}, \underline{r}) = \sum_{k=1}^{K} \log \left(f_{(y_k, z_k)}(x_k) \right). \tag{14}$$

To obtain MLE for λ_i we need to find the maximum of (14) by solving

$$\frac{\partial}{\partial \lambda_i} \sum_{k=1}^{K} I_{\{y_k=i\}} \log \left(\sum_{n=1}^{r_i} I_{\{z_k=n\}} f_{(i,n)}(x_k) \right) = 0, \tag{15}$$

that gives

$$\hat{\lambda}_i = \frac{\sum_{k=1}^{K} \sum_{n=1}^{r_i} n \cdot I_{\{y_k=i, z_k=n\}}}{\sum_{k=1}^{K} x_k I_{\{y_k=i\}}}. \tag{16}$$

To obtain MLE for probabilities $\hat{p}_{i,(j,m)}^{(c)}$ we apply [1] yielding

$$\hat{p}_{i,(j,m)}^{(c)} = \frac{\sum_{k=1}^{K-1} I_{\{c_k=c, y_k=i, y_{k+1}=j, z_{k+1}=m\}}}{\sum_{k=1}^{K-1} I_{\{y_k=i\}}}. \tag{17}$$

In the *E-step* of the EM algorithm the expected values of the unobserved variables are computed:

$$
\begin{aligned}
q_{(i,n),(j,m)}^{(c)}[k] &= P(c_k=c, y_k=i, z_k=n, y_{k+1}=j, z_{k+1}=m | \hat{\Theta}(\underline{r}), \mathcal{X}) \\
&= \frac{P(c_k=c, y_k=i, z_k=n, y_{k+1}=j, z_{k+1}=m, \mathcal{X} | \hat{\Theta}(\underline{r}))}{P(\mathcal{X} | \hat{\Theta}(\underline{r}))} \\
&= \frac{a_{(i,n)}[k-1] \cdot f_{(i,n)}(x_k) \cdot u_{(i,n),(j,m)}^{(c)} \cdot b_{(j,m)}[k+1]}{P(\mathcal{X} | \hat{\Theta}(\underline{r}))},
\end{aligned}
\tag{18}
$$

$$q_{(i,n)}[k] = \frac{P(y_k=i, z_k=n, \mathcal{X} | \hat{\Theta}(\underline{r}))}{P(\mathcal{X} | \hat{\Theta}(\underline{r}))} = \frac{a_{(i,n)}[k-1] \cdot b_{(i,n)}[k]}{P(\mathcal{X} | \hat{\Theta}(\underline{r}))}, \tag{19}$$

where $\hat{\Theta}(\underline{r})$ are the estimates of the parameters $\Theta(\underline{r})$.

Algorithm 1 Pseudo-code of the proposed EM algorithm

1: **procedure** HEM-FIT($x_k, \lambda_i, r_i, p_{i,(j,m)}^{(c)}$)
2:     $LogLi \leftarrow -\infty$
3:     **while** $(LogLi - oLogLi)/LogLi > \epsilon$ **do**
4:         Obtain vector $\underline{\pi}$ by (5)
5:         **for** $k = 1$ **to** K **do**
6:             Compute and store conditional densities $f_{(i,n)}(x_k)$ by (6)
7:         **end for**
8:         **for** $k = 0$ **to** K **do**
9:             Compute and store forward likelihood vectors $\underline{a}[k]$ by (12)
10:         **end for**
11:         **for** $k = K$ **downto** 1 **do**
12:             Compute and store backward likelihood vectors $\underline{b}[k]$ by (13)
13:         **end for**
14:         **for** $i = 1$ **to** M **do**
15:             Compute new estimate for λ_i by (20)
16:         **end for**
17:         **for** $c = 1$ **to** C, $i = 1$ **to** M, $j = 1$ **to** M **do**
18:             **for** $m = 1$ **to** r_j **do**
19:                 Compute new estimate for $p_{i,(j,m)}^{(c)}$ by (21)
20:             **end for**
21:         **end for**
22:         $oLogLi \leftarrow LogLi$
23:         $LogLi \leftarrow \underline{\pi} \cdot \underline{b}[0]$
24:     **end while**
25:     **return** $(\lambda_i, p_{i,(j,m)}^{(c)}), i, j = 1, \ldots, M, m = 1, \ldots, r_j, c = 1, \ldots, C)$
26: **end procedure**

In the *M-step* the new estimates of Θ (denoted by $\hat{\Theta}(\underline{r})$) are computed which maximize the expected likelihood function. From (17) it follows that

$$\hat{\lambda}_i = \frac{\sum_{k=1}^{K}\sum_{n=1}^{r_i} n \cdot q_{(i,n)}[k]}{\sum_{k=1}^{K} x_k \sum_{n=1}^{r_i} q_{(i,n)}[k]} = \frac{\sum_{k=1}^{K}\sum_{n=1}^{r_i} n \cdot a_{(i,n)}[k-1] \cdot b_{(i,n)}[k]}{\sum_{k=1}^{K} x_k \sum_{n=1}^{r_i} a_{(i,n)}[k-1] \cdot b_{(i,n)}[k]}, \quad (20)$$

$$\hat{p}_{i,(j,m)}^{(c)} = \frac{\sum_{k=1}^{K-1}\sum_{n=1}^{r_i} q_{(i,n),(j,m)}^{(c)}[k]}{\sum_{k=1}^{K-1}\sum_{n=1}^{r_i} q_{(i,n)}[k]}$$

$$= \frac{\sum_{k=1}^{K-1}\sum_{n=1}^{r_i} a_{(i,n)}[k-1] \cdot f_{(i,n)}(x_k) \cdot u_{(i,n),(j,m)}^{(c)} \cdot b_{(j,m)}[k+1]}{\sum_{k=1}^{K-1}\sum_{n=1}^{r_i} a_{(i,n)}[k-1] \cdot b_{(i,n)}[k]}. \quad (21)$$

The pseudo-code of the EM algorithm is depicted in Algorithm 1. The input of the algorithm is the trace and the initial guesses for probabilities $p_{i,(j,m)}^{(c)}$ and the shape and rate parameters of the branches. The outputs of the algorithm are the optimized values of these parameters.

For the initial guesses we apply the k-means algorithm as suggested in [16].

4.4 Optimization of the shape parameters

To make the algorithm more user friendly, it is possible to optimize the shape parameter vector $\underline{r} = (r_1, \ldots, r_M)$ as well. In this case the user has to enter just a single parameter: the size of the MAP he/she wants (N). The problem to determine the parameter N is addressed by statistical argument such as information criterion, but this is out of scope of this report.

The set of possible $\underline{r}$ vectors is given by

$$\mathcal{H}_N = \left\{ (r_1, \ldots, r_M); \sum_{m=1}^{M} r_i = N, 1 \leq M \leq N, 1 \leq r_1 \leq \cdots \leq r_M \right\}, \quad (22)$$

Note that the last condition $1 \leq r_1 \leq \cdots \leq r_M$ is based on the fact that each pair of any two values r_i and r_j is commutative; for example, $\underline{r} = (1, 2, 1)$ and $\underline{r} = (1, 1, 2)$ are supposed to be same.

Then the maximum likelihood estimates (MLEs) of shape parameters are given by the solution of the following maximization problem:

$$\hat{\underline{r}} = \underset{\underline{r} \in \mathcal{H}_N}{\operatorname{argmax}} L(\underline{r}, \hat{\Theta}(\underline{r}) | \mathcal{X}), \quad (23)$$

Since the shape parameters are restricted to an integer, (23) is essentially an integer programming, i.e., the combinational problem over a set $\mathcal{H}_N$.

A possible straight-forward solution of this optimization problem is to execute the presented EM algorithm with *all possible shape parameter vectors* $\underline{r} \in \mathcal{H}_N$, which can make the fitting very slow, since the cardinality of $\mathcal{H}_N$ increases exponentially as N grows.

Thummler et al. [20] presented a heuristic method that tries to predict promising combinations of shape parameters by doing a few iterations of the EM algorithm which is called the *progressive preselection*. This method first considers all vectors $\underline{r} \in \mathcal{H}_N$. Then the EM algorithm is started for each of the vectors with loose convergence conditions ($\epsilon = 10^{-2}$, usually requiring only a few iterations). The results are sorted according to the likelihood values, and half of the vectors (the worst performing ones) are dropped. Then further EM iterations are applied with the remaining vectors with tighter convergence conditions, the worst performing ones are dropped, and so on, till only a single vector $\underline{r}$ remains. While this approach is fast, it does not guarantee that it finds the optimal $\underline{r}$ at the end. The reason is that the optimal $\underline{r}$ can be dropped during the pre-selection steps if it converges slower to the optimum, which happened frequently in our numerical investigations if the convergence conditions corresponding to the preselection phases are not set adequately.

This paper presents an alternative approach to find the best combination of shape parameters with an *incremental approach*. The idea behind the method is to search only the neighborhood of a shape parameter vector. For a given sum of shape parameters N, we consider the following sets:

$$\tilde{\mathcal{H}}_k = \left\{ (r_1, \ldots, r_N); \sum_{m=1}^{N} r_i = k, 0 \leq r_1 \leq \cdots \leq r_N \right\}, \quad k = 1, \ldots, N. \quad (24)$$

The set $\tilde{\mathcal{H}}_k$ is essentially same as $\mathcal{H}_k$ in Eq. (22). Note that r_i is allowed to be 0 in $\tilde{\mathcal{H}}_k$ and that the length of all the elements of $\tilde{\mathcal{H}}_k$, $k = 1, \ldots, N$, becomes N. For all the elements in $\bigcup_{k=1}^{N} \tilde{\mathcal{H}}_k$, we define the following distance:

$$D(\underline{r}_i, \underline{r}_j) = \sum_{n=1}^{N} |r_{i,n} - r_{j,n}|, \quad \underline{r}_i, \underline{r}_j \in \bigcup_{k=1}^{N} \tilde{\mathcal{H}}_k, \tag{25}$$

where $\underline{r}_i = (r_{i,1}, \ldots, r_{i,N})$ and $\underline{r}_j = (r_{j,1}, \ldots, r_{j,N})$. According to the above distance, neighborhood of $\underline{r}$ is defined as follows.

$$\mathcal{N}(\underline{r}) = \left\{ \underline{r}'; D(\underline{r}, \underline{r}') = 1, \ \underline{r}' \in \bigcup_{k=1}^{N} \tilde{\mathcal{H}}_k \right\}. \tag{26}$$

For instance, if $N = 4$ the neighborhood of the vector $\underline{r} = (0, 0, 1, 2)$ is

$$\mathcal{N}(\underline{r}) = \{(0, 0, 0, 2), (0, 0, 1, 1), (0, 1, 1, 2), (0, 0, 2, 2), (0, 0, 1, 3)\}. \tag{27}$$

Observe that if $\underline{r} \in \tilde{\mathcal{H}}_k$, then a member of $\mathcal{N}(\underline{r})$ is a member of either $\tilde{\mathcal{H}}_{k-1}$ or $\tilde{\mathcal{H}}_{k+1}$.

Based on the above insights, we propose an algorithm based on the local search to find the MLE of shape parameter vector in Algorithm 2. In the algorithm, $L(\underline{r})$ means the log-likelihood function of $\underline{r}$. From the argument of degree of freedom in statistics, the maximum of log-likelihood functions in $\tilde{\mathcal{H}}_k$ is a non-decreasing function with respect to k. Then the algorithm searches for the maximum value in neighborhood with the direction from which k increases, and provides the (local) maximum of log-likelihood functions in the set $\tilde{\mathcal{H}}_N$. If the assumption that the neighborhood of the maximum of log-likelihood functions in $\tilde{\mathcal{H}}_k$ includes the shape parameter vector maximizing the log-likelihood function in $\tilde{\mathcal{H}}_{k-1}$ holds, the algorithm finds the global maximum of log-likelihood functions in $\tilde{\mathcal{H}}_N$. Although it is difficult to prove that this assumption holds for any situation, the assumption is expected to hold for many practical situation. The time complexity of Algorithm 2 is $O(N^{1.5})$, because the maximum size of neighborhood of $\mathcal{N}(\underline{r})$, $\underline{r} \in \tilde{\mathcal{H}}_k$ is proportional to a square of k. Since the size of $\mathcal{H}_N$ is given by a function of the factorial of N, the proposed method is applicable even for a large N.

5 Numerical Experiments

In this section we present two numerical examples to examine how effective our enhanced EM algorithm is.

5.1 Fitting Single-Class Trace

In the first example we intend to fit a well-known traffic trace, the BC-pAug89 trace[5] that is frequently used as a benchmark in several papers. It consist of the

---

[5] Downloaded from http://ita.ee.lbl.gov/html/contrib/BC.html

Algorithm 2 Incremental search for the best shape parameters.

1: $\underline{r} \leftarrow (0, \ldots, 0, 1)$
2: **for** $k = 2 : N$ **do**
3:      $L_{max} \leftarrow -\infty$
4:      **for** $\underline{r}' \in \mathcal{N}(\underline{r}) \bigcap \tilde{\mathcal{H}}_k$ **do**
5:          **if** $L(\underline{r}') > L_{max}$ **then**
6:              $\underline{r}_{max} \leftarrow \underline{r}'$
7:              $L_{max} \leftarrow L(\underline{r}')$
8:          **end if**
9:      **end for**
10:      $\underline{r} \leftarrow \underline{r}_{max}$
11: **end for**

---

inter-arrival times of one million packet arrivals measured on an Ethernet network. This trace does not distinguish multiple arrival types, thus we are applying single-class MAP fitting in this example. We are investigating two questions:

- How capable the proposed special MAP structure is when fitting the trace with the EM algorithm.
- How efficient the proposed heuristic method called "Incremental" is in optimizing the shape parameters $r_1, \ldots, r_M$.

The following MAP fitting methods are involved into the comparison.

- The EM-algorithm introduced in [5][6].
- The EM-algorithm published in [16] which operates on the general class of MAPs.
- The EM-algorithm operating on the ER-CHMM structure ([16]).
- The EM-algorithm operating on the special MAP structure proposed by this paper.

The latter two procedures are included in our new MAP/MMAP fitting tool called SPEM-FIT[7]. This open-source tool has been implemented in C++ and is able to utilize the multiple cores of modern CPUs. It supports three different methods to optimize the shape parameters, namely "progressive preselection" (referred to as "PreSel" in the sequel), enumerating all possible configurations and selecting the best one (referred to as "All"), and the new incremental method proposed in this paper ("Incr").

The likelihood values obtained when fitting with MAPs of different sizes are depicted in Figure 1. (The likelihood values in this section are all log-likelihoods divided by the length of the trace). The corresponding execution times are shown in Figure 2. We note that method "Buchholz" and "Okamura-Dohi" stopped before convergence when the maximum number of iterations has been reached (that is 1000 for "Buchholz", and 3000 for "Okamura-Dohi").

---

[6] We would like to thank Peter Buchholz and Jan Kriege for providing the implementation of the algorithm and for guidance on the usage
[7] It can be downloaded from https://bitbucket.org/ghorvath78/spemfit

When examining the execution times it is striking how slow the methods operating on the general MAP class are. Observe that these methods pick a single initial guess and apply several EM iterations on it, while in case of ER-CHMM and the proposed structure the execution times include the optimization of the shape parameters as well, they are still 1, 2, or even 3 orders of magnitudes faster. While being faster, they are also able to achieve as high or higher likelihood values than the EM algorithms working with general MAPs.

Table 1. Log-likelihood values obtained with different methods

Method	4	6	8	10	12
Buchhoz	-0.8033	-0.7957	-0.7929	n/a	n/a
Okamura-Dohi	-0.8002	-0.76345	-0.75512	-0.728709	-0.725897
ER-CHMM (All)	-0.80079	-0.77634	-0.74766	-0.72598	-0.715183
ER-CHMM (PreSel)	-0.800787	-0.776342	-0.747659	-0.723873	-0.715183
ER-CHMM (Incr)	-0.80079	-0.77936	-0.74766	-0.7314	-0.715183
Our (All)	-0.80051	-0.76494	-0.74173	-0.72813	-0.71399
Our (PreSel)	-0.800506	-0.764944	-0.741738	-0.727968	-0.713967
Our (Incr)	-0.80051	-0.76494	-0.74173	-0.72813	-0.71402

Based on the numerical results it is possible evaluate how the different components of our refined EM algorithm perform. With the proposed MAP structure it is possible to obtain higher likelihood values in most of the cases, although the execution time increases as well (which is straight forward as it has more parameters to optimize than the ER-CHMM). Regarding the optimization of the shape parameters we found that the "Incremental" procedure was able to find the optimum in the majority of the cases. With a small number of states "All" turned out to be faster than "Incremental", but from 8 states on "Incremental" catches up and the its speed advantage grows with increasing number of states.

Table 2. Fitting times obtained with different methods (in seconds)

Method	4	6	8	10	12
Buchhoz	33571	57703	103162	n/a	n/a
Okamura-Dohi	11172	17407	32896	60726	82773
ER-CHMM (All)	24	213	388	795	1954
ER-CHMM (PreSel)	25	94	113	337	582
ER-CHMM (Incr)	34	103	331	596	867
Our (All)	98	247	1026	2434	6179
Our (PreSel)	91	234	328	982	2257
Our (Incr)	139	450	1040	1988	3415

Regarding the "PreSel" heuristic, that does not guarantee finding the optimum, performs very well in this example. We note however, that fine-tuning the thresholds used by this method is a hard task. If the thresholds are too loose,

the procedure will be fast but it may drop candidates prematurely, potentially loosing the one that could provide the best result at the end. At the other hand, too tight thresholds make the progressive pre-selection practically equivalent to the "All" method. We selected the appropriate thresholds based on a large number of numerical experiments with several traces, however, these thresholds are not universal.

5.2 Fitting Multi-Class Trace

To examine the behavior of the proposed enhancements further, we made a multi-class trace from the BC-pAug89 trace according to the packet sizes. Arrivals with packet sizes between 1 and 759 are marked as class-1 arrivals, while arrivals of larger packets are considered as class-2 arrivals. As we do not have the implementations of EM algorithms for MMAPs published in the past, we compare the multi-class generalization of the ER-CHMM structure (our procedure with $p_{i,(j,m)}^{(c)} = 0$ for $m > 1$) and the proposed more general MMAP structure in this Section.

Table 3. Log-likelihood values obtained by using the two special structures

Vector r	ER-CHMM	Proposed	Vector r	ER-CHMM	Proposed
(1, 7)	-1.29603	-1.35266	(1, 1, 2, 4)	-1.00136	-0.997126
(2, 6)	-1.51657	-1.21412	(1, 1, 3, 3)	-1.02608	-0.997745
(3, 5)	-1.70301	-1.21671	(1, 2, 2, 3)	-1.03497	-1.10369
(4, 4)	-1.72712	-1.51611	(2, 2, 2, 2)	-1.05739	-1.04147
(1, 1, 6)	-1.11458	-1.11355	(1, 1, 1, 1, 4)	-1.05193	-1.05193
(1, 2, 5)	-1.18571	-1.12161	(1, 1, 1, 2, 3)	-1.01446	-1.09347
(1, 3, 4)	-1.21068	-1.15472	(1, 1, 2, 2, 2)	-1.05179	-1.03403
(2, 2, 4)	-1.19233	-1.17741	(1, 1, 1, 1, 1, 3)	-1.0743	-1.06969
(2, 3, 3)	-1.23412	-1.13608	(1, 1, 1, 1, 2, 2)	-1.04512	-1.09436
(1, 1, 1, 5)	-1.04259	-1.11015	(1, 1, 1, 1, 1, 1, 2)	-1.11112	-1.1638

Table 3 shows the log-likelihood values obtained by using the two special structures with all possible shape parameter configurations providing 8 states in total. Examining the results we can observe that the more general structure is able to achieve a significant improvement if there are several branches having a high shape parameter. If the shape parameter is low in most of the branches (or in the dominating branches having high steady state probability), the ER-CHMM performs better, as it has fewer parameters to optimize. In this particular example the optimal shape parameter vector is $(1, 1, 2, 4)$, where the proposed new structure wins by a slight margin.

Finally, Figure 1 compares the best results obtained with various MMAP sizes, showing that the EM algorithm with the proposed structure provides slightly better log-likelihood values, but the difference is marginal. With other traces, where the optimal shape parameters are higher, we expect the difference to be higher.

Fig. 1. Log-likelihood values obtained with different MMAP sizes
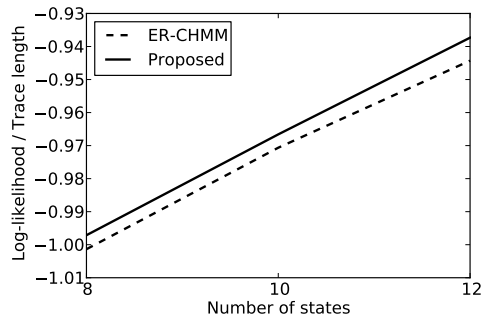
6 Conclusion

This paper presents several improvements on the EM algorithm based MAP fitting methods. As recognized in numerous past papers, the key idea to make MAP fitting efficient is to apply a special MAP structure instead of the general MAP class. We followed the same path in this paper. We generalized the ER-CHMM structure introduced in [16], developed a method to find the optimal shape parameters of this structure, finally, our method is able to fit multi-class traces as well. The proposed MAP structure is able to improve the log-likelihood values of the results of fitting in most of the cases investigated in the numerical experiments. The gain depends on the nature of the trace. In our numerical example the improvement is slight, but, since the execution time of the fitting is reasonable, it is still worth taking the advantage of this more general structure. With the procedure introduced to find the optimal shape parameters it is possible to avoid trying out all possible combinations, which is beneficial on the fitting times as well.

Acknowledgment

References

1. Theodore W. Anderson and Leo A. Goodman. Statistical inference about Markov chains. *The Annals of Mathematical Statistics*, pages 89–110, 1957.
2. Søren Asmussen, Olle Nerman, and Marita Olsson. Fitting phase-type distributions via the EM algorithm. *Scandinavian Journal of Statistics*, pages 419–441, 1996.
3. Falko Bause and Gábor Horváth. Fitting Markovian Arrival Processes by Incorporating Correlation into Phase Type Renewal Processes. In *Quantitative Evaluation*

of Systems (QEST), 2010 Seventh International Conference on the, pages 97–106. IEEE, 2010.

4. Lothar Breuer. An EM algorithm for batch Markovian arrival processes and its comparison to a simpler estimation procedure. *Annals of Operations Research*, 112(1-4):123–138, 2002.

5. Peter Buchholz. An EM-algorithm for MAP fitting from real traffic data. In *Computer Performance Evaluation. Modelling Techniques and Tools*, pages 218–236. Springer, 2003.

6. Peter Buchholz, Peter Kemper, and Jan Kriege. Multi-class Markovian arrival processes and their parameter fitting. *Performance Evaluation*, 67(11):1092–1106, 2010.

7. Peter Buchholz and Jan Kriege. A heuristic approach for fitting MAPs to moments and joint moments. In *Quantitative Evaluation of Systems, 2009. QEST'09. Sixth International Conference on the*, pages 53–62. IEEE, 2009.

8. Giuliano Casale, Eddy Z. Zhang, and Evgenia Smirni. KPC-toolbox: Simple yet effective trace fitting using Markovian arrival processes. In *Quantitative Evaluation of Systems, 2008. QEST'08. Fifth International Conference on*, pages 83–92. IEEE, 2008.

9. Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38, 1977.

10. Rachid El Abdouni Khayari, Ramin Sadre, and Boudewijn R Haverkort. Fitting world-wide web request traces with the EM-algorithm. *Performance Evaluation*, 52(2):175–191, 2003.

11. Q. He and M.F. Neuts. Markov chains with marked transitions. *Stochastic Processes and their Applications*, 74:37–52, 1998.

12. András Horváth, Gábor Horváth, and Miklós Telek. A traffic based decomposition of two-class queueing networks with priority service. *Computer Networks*, 53(8):1235–1248, 2009.

13. Gábor Horváth, Peter Buchholz, and Miklós Telek. A MAP fitting approach with independent approximation of the inter-arrival time distribution and the lag correlation. In *Quantitative Evaluation of Systems, 2005. Second International Conference on the*, pages 124–133. IEEE, 2005.

14. Alexander Klemm, Christoph Lindemann, and Marco Lohmann. Modeling IP traffic using the batch Markovian arrival process. *Performance Evaluation*, 54(2):149–173, 2003.

15. Guy Latouche and Vaidyanathan Ramaswami. *Introduction to matrix analytic methods in stochastic modeling*, volume 5. Society for Industrial and Applied Mathematics, 1987.

16. Hiroyuki Okamura and Tadashi Dohi. Faster maximum likelihood estimation algorithms for Markovian arrival processes. In *Quantitative Evaluation of Systems, 2009. QEST'09. Sixth International Conference on the*, pages 73–82. IEEE, 2009.

17. Hiroyuki Okamura, Tadashi Dohi, and Kishor S Trivedi. A refined EM algorithm for PH distributions. *Performance Evaluation*, 68(10):938–954, 2011.

18. T. Rydén. An EM algorithm for estimation in Markov-modulated Poisson processes. *Computational Statistics & Data Analysis*, 21(4):431–447, 1996.

19. M. Telek and G. Horváth. A minimal representation of Markov arrival processes and a moments matching method. *Performance Evaluation*, 64(9):1153–1168, 2007.

20. Axel Thummler, Peter Buchholz, and Miklos Telek. A novel approach for phase-type fitting with the EM algorithm. *Dependable and Secure Computing, IEEE Transactions on*, 3(3):245–258, 2006.