PH and MAP Fitting with Aggregated Traffic Traces

Jan Kriege, Peter Buchholz
Department of Computer Science, TU Dortmund,
44221 Dortmund, Germany
{jan.kriege,peter.buchholz}@udo.edu

Author prepared version of a paper published in

Measurement, Modelling, and Evaluation of Computing Systems and Dependability and Fault Tolerance

Proceedings of the MMB & DFT 2014

Lecture Notes in Computer Science Volume 8376, 2014

The original publication is available at www.springerlink.com

http://dx.doi.org/10.1007/978-3-319-05359-2_1

PH and MAP Fitting with Aggregated Traffic Traces

Jan Kriege, Peter Buchholz
Department of Computer Science, TU Dortmund,
44221 Dortmund, Germany
{jan.kriege,peter.buchholz}@udo.edu

Abstract

Phase Type Distributions (PHDs) and *Markovian Arrival Processes* (MAPs) are versatile models for the modeling of timing behavior in stochastic models. The parameterization of the models according to measured traces is often done using *Expectation Maximization* (EM) algorithms, which have long runtimes when applied to realistic datasets. In this paper, new versions of EM algorithms are presented that use only an aggregated version of the trace. Experiments show that these realizations of EM algorithms are much more efficient than available EM algorithms working on the complete trace and the fitting quality remains more or less the same.

Keywords: Phase-type distributions, Markovian Arrival Processes, Expectation Maximization Algorithm, Trace Aggregation

1 Introduction

An adequate representation of inter-event times, let it be inter-arrival times of packets in a computer network, time between failure in dependable systems, or service times in computer systems, is a fundamental requirement in stochastic modeling. For models that are solved using analytical or numerical techniques and often also for simulation models inter-event times are described by Markov models. Based on the work of Marcel Neuts [20, 21], *Phase Type Distributions* (PHDs) and *Markovian Arrival Processes* (MAPs) have been defined and are nowadays widely used in stochastic modeling. PHDs and MAPs are quite general classes of stochastic models that allow one to approximate many distributions and stochastic processes arbitrarily closely [22, 11]. However, the parameterization of the models to match some observed behavior is a complex non-linear optimization problem. The general problem is to find the parameters of a PHD or MAP of a given order with respect to an observed sequence of inter-event times such that it is likely that the observed sequence is generated from the PHD or MAP, respectively. This is commonly denoted as parameter fitting. In some application areas, like computer networks, traces can be huge and contain more than a million entries. Methods for parameter fitting of PHDs are a research topic for more than 30 years. Parameter fitting for MAPs is much harder such that most available methods have been developed in the last 10 years. The number of available papers on parameter fitting of PHDs or MAPs is much too large to mention only a significant portion of them here. Therefore we present only the most important results for the algorithms that are developed in this paper. Overviews on available fitting methods can be found in [1, 5, 10].

Parameter fitting for PHDs can be made according to the whole sequence of measured data or based on some derived measures, like some moments. Moment based fitting methods can be found for example in [7, 12, 14]. Moment based fitting is often efficient and almost independent of the number of elements in the trace because only the empirical moments have to be computed from the trace. Fitting based on the first two or three moments is fairly easy. However, when increasing the number of moments usually two problems arise. First, it is sometimes hard to match all moments together with a PHD of given order, such that approximation methods have to be applied [6]. Additionally, estimators for higher order moments are unreliable. As shown in [17], confidence intervals for the moments derived from network traces with more than a million entries are extremely wide if the order of the moments is beyond 3 or 4. This implies that moment based parameter fitting usually has a limited accuracy since the shape of the density is not adequately described by lower order moments. Moment based fitting can be extended to MAPs [6, 8, 13] by using joint moments or some lags of the coefficient of autocorrelation. However, the problems of the approach are similar to those of moment based fitting approaches for PHDs. Higher order joint moments are hard to approximate and lower order joint moments do not adequately capture the correlation structure. Based on the complete trace data, parameters of a PHD or MAP can be fitted to approximate the empirical density. Most adequate for this purpose are *Expectation Maximization* (EM) algorithms. The first EM algorithm for PHDs has been proposed by Asmussen and his coworkers [2]. Later extensions of the approach improve the runtimes significantly by using more efficient computations and by restricting the classes of PHDs to acyclic structures [9, 15, 24, 29]. The use of acyclic PHDs usually does not reduce the fitting quality. EM algorithms outperform moment based fitting approaches for multimodal densities. EM algorithms are iterative algorithms which are guaranteed to find local optima. However, convergence can be very slow and the time per iteration depends linearly on the number of elements in the trace, the number of parameters (i.e., the order of the PHD) and the difference between small and large elements in the trace. Especially for large network traces fitting times can be prohibitive. In [26] methods are introduced to aggregate the values in a trace and apply afterwards the EM algorithm for the aggregated trace. Examples show that the aggregation step often reduces the fitting times by orders of a magnitude and results in a similar likelihood value, which is a measurement for the fitting quality.

EM algorithms for parameter fitting of MAPs have also been published [3, 4, 16]. The algorithms work well for small traces, in particular, if these traces have been generated from MAPs of a lower order. If the algorithms are applied to huge network traces, then the convergence is slow and fitting times are often extremely long [17]. This seriously restricts the usability of EM algorithms for MAP fitting of many real problems in modeling computer networks.

In this paper we improve EM algorithms for PHD and MAP fitting. This is done by some slight modifications in the basic EM algorithms and more important by extending the trace aggregation of [26] from PHDs to MAPs. The original approach cannot be used for MAPs because after aggregation, dependencies between elements in the trace are no longer available. Here we extend the approach by keeping some of the information on dependencies in the aggregated trace and use this in the EM algorithm to fit the MAP matrices. Examples show that the aggregated information is sufficient to describe the dependencies in the trace and that the likelihood value of a MAP resulting from an aggregated trace is as good as the likelihood value of a MAP that has been fitted using the original trace. However, fitting based on the aggregated trace is often orders of magnitude faster than the use of standard EM algorithms for the complete trace.

The paper is structured as follows. In the next section the definitions and notations are introduced and the basic EM algorithms for PHD and MAP fitting are presented. Section 3 introduces an aggregation of traces that keeps dependencies partially in the aggregated trace and the extension of EM algorithms for aggregated traces. Afterwards, in Section 4, the new algorithms are applied to one synthetic and two real traces. The paper ends with the conclusions.

2 Basic Definitions

We present the basic notations and definitions which are used in the paper. For further details on PHDs and MAPs we refer to the literature [5]. Furthermore, we introduce a basic EM algorithm which is derived from the EM algorithms proposed in the literature [3, 4, 16, 24].

2.1 Input Modeling

Input modeling is one of the major steps when building stochastic models for real world systems. It is used to define the parameters of a stochastic model, in particular to define the duration of events or the time between events. Usually this modeling is based on some measurements taken from a real world system. These measurements will be denoted as a trace $\mathcal{T} = (t_1, \ldots, t_m)$ where $t_i \geq 0$ describes the ith observation. Observations can be times between events (e.g., the arrival of packets in a computer networks, the occurrence of failures) or the size or duration of an event (e.g., the packet size in computer networks, the duration of a repair operation, the service time at server). We usually use the term inter-event times, even if this does not completely capture all situations. m is the order of the trace which can be large, in particular if it results from some automated measurement as used for example in computer networks. From a trace, standard measures like the moments, joint moments, the empirical density or empirical distribution function can be derived using standard means [5]. The goal is to find a stochastic model such that it is likely that the trace has been generated as an output of the model. As model types we consider PHDs for uncorrelated data and MAPs for correlated data. Both model types are introduced in the following paragraphs.

2.2 Phase-Type Distributions

A phase type distribution (PHD) can be interpreted as an absorbing Markov chain with n transient and 1 absorbing state and generator

$$\boldsymbol{Q} = \begin{pmatrix} \boldsymbol{D}_0 & \boldsymbol{t} \\ \mathbf{0} & 0 \end{pmatrix}$$

where $\boldsymbol{t} = -\boldsymbol{D}_0 \mathbb{1}$ and $\boldsymbol{D}_0$ is a non-singular $n \times n$ matrix with non-negative non-diagonal elements and row sums smaller or equal to 0. We assume that the initial vector of the PHD equals $(\boldsymbol{\pi}, 0)$ which means that the distribution has no point mass at zero. Then the PHD can be characterized by $(\boldsymbol{\pi}, \boldsymbol{D}_0)$. Moments, density and distribution function can be computed from this representation using standard means [5, 17]. If the parameters of a PHD have been computed to match the values of some trace $\mathcal{T}$, then the fitting quality can be measured in terms of the likelihood or log likelihood value which is computed as follows.

$$\mathcal{L} = \prod_{i=1}^{m} \boldsymbol{\pi} e^{\boldsymbol{D}_0 t_i} \boldsymbol{t} \text{ or } \log \mathcal{L} = \sum_{i=1}^{m} \log \left(\boldsymbol{\pi} e^{\boldsymbol{D}_0 t_i} \boldsymbol{t} \right). \tag{1}$$

The better the likelihood value, the better the fitting quality, but for a given trace it is not clear which likelihood value can be or should be achieved.

2.3 Markovian Arrival Processes

PHDs can be used to describe identically and independently distributed values. In many real applications, for example the inter-arrival times at a router in a computer network, times are strongly correlated such that the modeling with PHDs is not adequate. For this purpose MAPs have been defined [20]. A MAP is defined by a pair of matrices $(\boldsymbol{D}_0, \boldsymbol{D}_1)$ such that $\boldsymbol{Q} = \boldsymbol{D}_0 + \boldsymbol{D}_1$ is an irreducible generator matrix, $\boldsymbol{D}_1$ is non-negative and $\boldsymbol{D}_0$ has negative diagonal and non-negative non-diagonal entries. We furthermore assume that $\boldsymbol{D}_0$ is non-singular to avoid degenerated inter-event time distributions. The behavior of a MAP is as follows. The process behaves like a Markov chain with generator matrix $\boldsymbol{Q}$ and whenever a transition from $\boldsymbol{D}_1$ occurs, an event is generated. Every PHD $(\boldsymbol{\pi}, \boldsymbol{D}_0)$ can be represented as an equivalent MAP $(\boldsymbol{D}_0, (-\boldsymbol{D}_0 \mathbb{1})\boldsymbol{\pi})$. MAPs allow one to model correlated inter-event times. For the computation of basic quantities like moments, joint moments or (joint) densities we refer again to the literature [5, 17].

The likelihood and log likelihood for a MAP according to a trace $\mathcal{T}$ are defined as

$$\mathcal{L} = \boldsymbol{\pi} \prod_{i=1}^{m} \left(e^{\boldsymbol{D}_0 t_i} \boldsymbol{D}_1 \right) \mathbb{1} \text{ and } \log \mathcal{L} = \sum_{i=1}^{m} \log \left(\boldsymbol{\pi}^i e^{\boldsymbol{D}_0 t_i} \boldsymbol{D}_1 \mathbb{1} \right) \tag{2}$$

where $\boldsymbol{\pi}^i = \boldsymbol{\pi} \prod_{j=1}^{i-1} \left(e^{\boldsymbol{D}_0 t_i} \boldsymbol{D}_1 \right)$ and $\boldsymbol{\pi}$ is an appropriately chosen initial vector. Often the embedded stationary distribution after an event, that is given by the solution of $\boldsymbol{\pi} \left(\boldsymbol{D}_0 \right)^{-1} \boldsymbol{D}_1 = \boldsymbol{\pi}$ subject to $\boldsymbol{\pi} \mathbb{1} = 1$, is used as initial vector.

2.4 EM Algorithms for PH and MAP Fitting

EM algorithms are local optimization algorithms that compute the parameters of some stochastic model from incomplete data. They are nowadays very popular in several branches of computer science and statistics. For the general idea behind the algorithms we refer to the literature [19]. Here we concentrate on specific realizations of EM algorithms for PHDs and MAPs.

An EM algorithm consists of an E-step that determines the expectation of some flows under the current parameter setting and an M-step that changes the parameter in such a way that the flows are maximized assuming the behavior that has been analyzed in the E-step. It can be shown that the combination of the two steps results in a non-decreasing sequence of flows and the flows determine the value of the likelihood function (Eqs. 1 or 2). In this way, a local optimizer is defined that computes the parameters of a PHD or MAP with a locally optimal value of the likelihood function.

We begin with the EM algorithm for PHDs and define the following vectors and matrix.

$$\begin{aligned} \boldsymbol{f}_{(\boldsymbol{\pi},\boldsymbol{D}_0),t} &= \boldsymbol{\pi} e^{\boldsymbol{D}_0 t}, \quad \boldsymbol{b}_{(\boldsymbol{\pi},\boldsymbol{D}_0),t} = e^{\boldsymbol{D}_0 t}\boldsymbol{t}, \\ \boldsymbol{F}_{(\boldsymbol{\pi},\boldsymbol{D}_0),t} &= \int_0^t \left(\boldsymbol{f}_{(\boldsymbol{\pi},\boldsymbol{D}_0),t-u}\right)^T \left(\boldsymbol{b}_{(\boldsymbol{\pi},\boldsymbol{D}_0),u}\right)^T du. \end{aligned} \tag{3}$$

Matrix $\boldsymbol{F}_{(\boldsymbol{\pi},\boldsymbol{D}_0),t}$ contains the flow between two states for the given PHD. The vectors in matrix from (3) can be computed using uniformization [28]. Let $\alpha \geq \max_i |\boldsymbol{D}_0(i,i)|$ and $\boldsymbol{P}_0 = \boldsymbol{D}_0/\alpha + \boldsymbol{I}$. We can define the following two sequences of vectors

$$\boldsymbol{v}^{(0)} = \boldsymbol{\pi} \text{ and } \quad \boldsymbol{v}^{(k+1)} = \boldsymbol{v}^{(k)}\boldsymbol{P}_0, \qquad \boldsymbol{w}^{(0)} = \boldsymbol{t} \text{ and } \quad \boldsymbol{w}^{(k+1)} = \boldsymbol{P}_0\boldsymbol{w}^{(k)} \tag{4}$$

for $k = 0, 1, 2, \ldots$ such that

$$\begin{array}{l} \boldsymbol{f}_{(\boldsymbol{\pi},\boldsymbol{D}_0),t} = \sum\limits_{k=0}^{\infty} \beta(\alpha t, k)\boldsymbol{v}^{(k)}, \quad \boldsymbol{b}_{(\boldsymbol{\pi},\boldsymbol{D}_0),t} = \sum\limits_{k=0}^{\infty} \beta(\alpha t, k)\boldsymbol{w}^{(k)} \\ \boldsymbol{F}_{(\boldsymbol{\pi},\boldsymbol{D}_0),t} = \frac{1}{\alpha}\sum\limits_{k=0}^{\infty} \beta(\alpha t, k+1)\sum\limits_{l=0}^{k} \left(\boldsymbol{v}^{(l)}\right)^T \left(\boldsymbol{w}^{(k-l)}\right)^T, \end{array} \tag{5}$$

where $\beta(\alpha t, k)$ is the probability of k events of a Poisson process with rate α in the interval $(0, t]$. For practical computations, the infinite summations have to be truncated which can be done such that predefined error bounds for the results are observed [28].

We now define some random variables. B_i is the number of times the PHD starts in state i, N_{ij} the number of times the PHD goes from transient state i to j, N_{in+1} the number of times an event is generated from state i and Z_i the total time spent in state i. All values are computed according to a given trace $\mathcal{T}$.

$$\begin{array}{llll} E_{(\boldsymbol{\pi},\boldsymbol{D}_0),\mathcal{T}}[B_i] & = \sum\limits_{k=1}^{m} \frac{\boldsymbol{\pi}(i)\boldsymbol{b}_{(\boldsymbol{\pi},\boldsymbol{D}_0),t_k}(i)}{\boldsymbol{\pi}\boldsymbol{b}_{(\boldsymbol{\pi},\boldsymbol{D}_0),t_k}}, & E_{(\boldsymbol{\pi},\boldsymbol{D}_0),\mathcal{T}}[Z_i] & = \sum\limits_{k=1}^{m} \frac{\boldsymbol{F}_{(\boldsymbol{\pi},\boldsymbol{D}_0),t_k}(i,i)}{\boldsymbol{\pi}\boldsymbol{b}_{(\boldsymbol{\pi},\boldsymbol{D}_0),t_k}}, \\ E_{(\boldsymbol{\pi},\boldsymbol{D}_0),\mathcal{T}}[N_{ij}] & = \sum\limits_{k=1}^{m} \frac{\boldsymbol{D}_0(i,j)\boldsymbol{F}_{(\boldsymbol{\pi},\boldsymbol{D}_0),t_k}(i,j)}{\boldsymbol{\pi}\boldsymbol{b}_{(\boldsymbol{\pi},\boldsymbol{D}_0),t_k}}, & E_{(\boldsymbol{\pi},\boldsymbol{D}_0),\mathcal{T}}[N_{in+1}] & = \sum\limits_{k=1}^{m} \frac{\boldsymbol{t}(i)\boldsymbol{f}_{(\boldsymbol{\pi},\boldsymbol{D}_0),t_k}(i)}{\boldsymbol{\pi}\boldsymbol{b}_{(\boldsymbol{\pi},\boldsymbol{D}_0),t_k}} \end{array} \tag{6}$$

The equation describes the E-step of the EM algorithm. The following M-step computes new estimates for the parameters from the results of the E-step.

$$\begin{array}{ll} \hat{\boldsymbol{\pi}}(i) = \frac{E_{(\boldsymbol{\pi},\boldsymbol{D}_0),\mathcal{T}}[B_i]}{m}, & \hat{\boldsymbol{D}}_0(i,j) = \frac{E_{(\boldsymbol{\pi},\boldsymbol{D}_0),\mathcal{T}}[N_{ij}]}{E_{(\boldsymbol{\pi},\boldsymbol{D}_0),\mathcal{T}}[Z_i]}, \\ \hat{\boldsymbol{t}}(i) = \frac{E_{(\boldsymbol{\pi},\boldsymbol{D}_0),\mathcal{T}}[N_{in+1}]}{E_{(\boldsymbol{\pi},\boldsymbol{D}_0),\mathcal{T}}[Z_i]}, & \hat{\boldsymbol{D}}_0(i,i) = -(\hat{\boldsymbol{t}}(i) + \sum\limits_{i \neq j}^{n} \hat{\boldsymbol{D}}_0(i,j)). \end{array} \tag{7}$$

The iteration between (6) and (7) until the parameters are stable defines an EM algorithm for PHDs.

For MAPs the algorithm works similar but requires more effort since it is not sufficient to consider only the isolated values in a trace, instead the whole sequence has to be used in the E-step. Define first the following two vectors.

$$\begin{aligned} \boldsymbol{ff}^{(k)}_{(\boldsymbol{D}_0,\boldsymbol{D}_1)} &= \begin{cases} \boldsymbol{\pi} & \text{if } k = 1 \\ \boldsymbol{ff}^{(k-1)}_{(\boldsymbol{D}_0,\boldsymbol{D}_1)} e^{\boldsymbol{D}_0 t_{k-1}} \boldsymbol{D}_1 & \text{if } 1 < k \leq m \end{cases} \\ \boldsymbol{bb}^{(k)}_{(\boldsymbol{D}_0,\boldsymbol{D}_1)} &= \begin{cases} \boldsymbol{D}_1 \mathbb{1} & \text{if } k = m \\ \boldsymbol{D}_1 e^{\boldsymbol{D}_0 t_{k+1}} \boldsymbol{bb}^{(k+1)}_{(\boldsymbol{D}_0,\boldsymbol{D}_1)} & \text{if } 1 \leq k < m \end{cases} \end{aligned} \tag{8}$$

Vector $\boldsymbol{ff}^{(k)}_{\cdot}$ is the forward vector describing the distribution immediately after the $(k-1)$th event in the trace and $\boldsymbol{bb}^{(k)}_{\cdot}$ is the backward vector immediately after the events $m, \ldots, k+1$ occurred in backward direction. With these vectors the flow vectors and flow matrix for a single event can be defined.

$$\begin{array}{l} \boldsymbol{f}^{(k)}_{(\boldsymbol{D}_0,\boldsymbol{D}_1),t} = \boldsymbol{ff}^{(k)}_{(\boldsymbol{D}_0,\boldsymbol{D}_1)} e^{\boldsymbol{D}_0 t},\ \boldsymbol{b}^{(k)}_{(\boldsymbol{D}_0,\boldsymbol{D}_1),t} = e^{\boldsymbol{D}_0 t} \boldsymbol{bb}^{(k)}_{(\boldsymbol{D}_0,\boldsymbol{D}_1)}, \text{ and} \\ \boldsymbol{F}^{(k)}_{(\boldsymbol{D}_0,\boldsymbol{D}_1),t} = \int_0^t \left(\boldsymbol{f}^{(k)}_{(\boldsymbol{D}_0,\boldsymbol{D}_1),t-u}\right)^T \left(\boldsymbol{b}^{(k)}_{(\boldsymbol{D}_0,\boldsymbol{D}_1),u}\right)^T du. \end{array} \tag{9}$$

For the E-step of a MAP, the random variable M_{ij} is introduced that defines the number of times a transitions from state i to state j occurs that is accompanied by an event.

$$\begin{array}{ll} E_{(\boldsymbol{D}_0,\boldsymbol{D}_1),\mathcal{T}}[Z_i] & = \sum\limits_{k=1}^{m} \frac{\boldsymbol{F}^{(k)}_{(\boldsymbol{D}_0,\boldsymbol{D}_1),t_k}(i,i)}{\boldsymbol{\pi b}^{(1)}_{(\boldsymbol{D}_0,\boldsymbol{D}_1),t_1}},\ E_{(\boldsymbol{D}_0,\boldsymbol{D}_1),\mathcal{T}}[N_{ij}] = \sum\limits_{k=1}^{m} \frac{\boldsymbol{D}_0(i,j)\boldsymbol{F}_{(\boldsymbol{D}_0,\boldsymbol{D}_1),t_k}(i,j)}{\boldsymbol{\pi b}^{(1)}_{(\boldsymbol{D}_0,\boldsymbol{D}_1),t_1}}, \\ E_{(\boldsymbol{D}_0,\boldsymbol{D}_1),\mathcal{T}}[M_{ij}] & = \sum\limits_{k=1}^{m-1} \frac{\boldsymbol{f}^{(k)}_{(\boldsymbol{D}_0,\boldsymbol{D}_1),t_k}(i)\boldsymbol{D}_1(i,j)\boldsymbol{b}^{(k+1)}_{(\boldsymbol{D}_0,\boldsymbol{D}_1),t_{k+1}}}{\boldsymbol{\pi b}^{(1)}_{(\boldsymbol{D}_0,\boldsymbol{D}_1),t_1}} + \frac{\boldsymbol{f}^{(m)}_{(\boldsymbol{D}_0,\boldsymbol{D}_1),t_m}(i)\boldsymbol{D}_1(i,j)\,\mathbb{1}}{\boldsymbol{\pi b}^{(1)}_{(\boldsymbol{D}_0,\boldsymbol{D}_1),t_1}} \end{array} \tag{10}$$

Then the M-step is given by

$$\begin{array}{ll} \hat{\boldsymbol{D}}_0(i,j) & = \frac{E_{(\boldsymbol{D}_0,\boldsymbol{D}_1),\mathcal{T}}[N_{ij}]}{E_{(\boldsymbol{D}_0,\boldsymbol{D}_1),\mathcal{T}}[Z_i]} \text{ for } i \neq j,\ \hat{\boldsymbol{D}}_1(i,j) = \frac{E_{(\boldsymbol{D}_0,\boldsymbol{D}_1),\mathcal{T}}[M_{ij}]}{E_{(\boldsymbol{D}_0,\boldsymbol{D}_1),\mathcal{T}}[Z_i]}, \\ \hat{\boldsymbol{D}}_0(i,i) & = -\left(\sum\limits_{j=1,i\neq j}^{n} \hat{\boldsymbol{D}}_0(i,j) + \sum\limits_{j=1}^{n} \hat{\boldsymbol{D}}_1(i,j)\right). \end{array} \tag{11}$$

Again the iteration of E-step (10) and M-step (11) defines the EM-algorithm.

3 An Expectation Maximization Algorithm using Aggregated Traffic Traces

An EM algorithm using trace aggregation for a subclass of PHDs, namely Hyper-Erlang distributions, has been introduced in [26]. [26] proposes two different aggregation methods that will be summarized in the following section and extended to account for the additional requirements when fitting MAPs. Afterwards we will outline an EM algorithm with trace aggregation for general PHDs and MAPs.

3.1 Trace Aggregation

Let $\mathcal{T} = (t_1, t_2, \cdots, t_m)$ be a trace of e.g. inter-event times t_i. For trace aggregation the interval $[\min(\mathcal{T}), \max(\mathcal{T})]$ is divided into M subintervals $(\Delta_0, \Delta_1]$, $(\Delta_1, \Delta_2], \cdots, (\Delta_{M-1}, \Delta_M]$ with $0 \leq \Delta_0 < \min(\mathcal{T}) < \Delta_1 < \cdots < \Delta_{M-1} < \Delta_M = \max(\mathcal{T})$. [26] presented two aggregation methods that differ in the way how the interval boundaries Δ_i are chosen. In the uniform aggregation approach all intervals have the same width $\Delta = \Delta_i - \Delta_{i-1} = (\max(\mathcal{T}) - \Delta_0)/M$. Each interval i is then represented by a tuple $(\hat{t}_i, w_i)$, where $\hat{t}_i$ is the mean value of elements that fall within this interval and the weight w_i corresponds to the number of elements in the interval. More formally, let $J_i, i = 1, \cdots, M$ be sets of indices such that $j \in J_i$ if $t_j \in (\Delta_{i-1}, \Delta_i]$. Then, $w_i = |J_i|$ and $\hat{t}_i = \frac{1}{w_i} \sum_{j \in J_i} t_j$.

If the empirical distribution of the trace is heavy-tailed, uniform trace aggregation will produce a few intervals with lots of elements and many intervals with only few or even no elements. In these cases the logarithmic trace aggregation described in [26] provides better results. For logarithmic trace aggregation the intervals are chosen with equidistant width on a logarithmic scale, e.g. $(10^{-3}, 10^{-2}]$, $(10^{-2}, 10^{-1}]$, $(10^{-1}, 10^{0}]$, $(10^{0}, 10^{1}], \cdots$. Let $s_{min} = \lfloor \log_{10} \min(\mathcal{T}) \rfloor$ and $s_{max} = \lceil \log_{10} \max(\mathcal{T}) \rceil$ be the smallest and largest logarithmic scale, respectively. Then, the trace is divided into the intervals $(10^{s}, 10^{s+1}], s = s_{min}, \cdots, s_{max} - 1$, i.e. we have for the i-th interval $(\Delta_{i-1}, \Delta_i] = (10^{s_{min}+i-1}, 10^{s_{min}+i}]$. It is also possible to further divide the resulting intervals using uniform trace aggregation. Again, each interval i is then represented by a tuple $(\hat{t}_i, w_i)$. Intervals with $w_i = 0$ can be ignored for both aggregation types.

The aggregation approaches from [26] described above can be applied for PHD fitting but are not suitable for MAP fitting, because the order of the t_i and thereby the information about the correlation is lost. To save this information we introduce the vector $\mathcal{S} = (s_1, s_2, \cdots, s_m)$ with $s_j = i$ if $t_j \in (\Delta_{i-1}, \Delta_i)$, i.e. we store for each trace element the number of its interval. Thus, the aggregated trace $\mathcal{T}^*$ consists of M tuples $(\hat{t}_i, w_i)$ and the sequence $\mathcal{S}$.

3.2 EM Algorithms for PHD and MAP Fitting with Trace Aggregation

In the following we will modify the algorithms from Sect. 2 to allow for PHD and MAP fitting with aggregated traffic traces. We will first outline an algorithm for general PHDs which is a generalization of the approach from [26]. Afterwards the algorithm is extended to the case where MAPs are considered.

Assume that an aggregated traffic trace $\mathcal{T}^*$ is given by M tuples $(\hat{t}_i, w_i)$ and the sequence $\mathcal{S}$. For the description of the algorithm it does not matter how the intervals have been obtained, however, for the experimental results in the next section it is, of course, important which aggregation method has been used. For fitting the PHD we will only use $(\hat{t}_i, w_i)$, for fitting the MAP also $\mathcal{S}$ is required.

3.2.1 Fitting PHDs:

The modifications on the EM algorithm for PHDs from Sect. 2 to allow for using aggregated traffic traces mostly affect the E-step of the algorithm.

In the original approach we have to compute the forward vector $\boldsymbol{f}_{(\boldsymbol{\pi},\boldsymbol{D}_0),t}$, the backward vector $\boldsymbol{b}_{(\boldsymbol{\pi},\boldsymbol{D}_0),t}$ and the flow matrix $\boldsymbol{F}_{(\boldsymbol{\pi},\boldsymbol{D}_0),t}$ (cf. Eq. 3) for each trace element $t = t_1, t_2, \cdots, t_m$. Since these values are computed independent of each other, Eq. 3 can also be used for the aggregated trace, i.e. we compute $\boldsymbol{f}_{(\boldsymbol{\pi},\boldsymbol{D}_0),t}$, $\boldsymbol{b}_{(\boldsymbol{\pi},\boldsymbol{D}_0),t}$ and $\boldsymbol{F}_{(\boldsymbol{\pi},\boldsymbol{D}_0),t}$ for each mean value of the intervals $t = \hat{t}_1, \hat{t}_2, \cdots, \hat{t}_M$. This leads to significant improvement of the runtime of the algorithm, because the time consuming computations of the matrix exponential and the integral in Eq. 3 have only to be done $M << m$ times.

In the E-step of the algorithm the weights w_i have to be considered, i.e. Eq. 6 becomes

$$\begin{aligned}
E_{(\boldsymbol{\pi},\boldsymbol{D}_0),\mathcal{T}^*}[B_i] &= \sum_{k=1}^{M} w_k \frac{\boldsymbol{\pi}(i)\boldsymbol{f}_{(\boldsymbol{\pi},\boldsymbol{D}_0),\hat{t}_k}(i)}{\boldsymbol{\pi}\boldsymbol{b}_{(\boldsymbol{\pi},\boldsymbol{D}_0),\hat{t}_k}} \\
E_{(\boldsymbol{\pi},\boldsymbol{D}_0),\mathcal{T}^*}[Z_i] &= \sum_{k=1}^{M} w_k \frac{\boldsymbol{F}_{(\boldsymbol{\pi},\boldsymbol{D}_0),\hat{t}_k}(i,i)}{\boldsymbol{\pi}\boldsymbol{b}_{(\boldsymbol{\pi},\boldsymbol{D}_0),\hat{t}_k}} \\
E_{(\boldsymbol{\pi},\boldsymbol{D}_0),\mathcal{T}^*}[N_{ij}] &= \sum_{k=1}^{M} w_k \frac{\boldsymbol{D}_0(i,j)\boldsymbol{F}_{(\boldsymbol{\pi},\boldsymbol{D}_0),\hat{t}_k}(i,j)}{\boldsymbol{\pi}\boldsymbol{b}_{(\boldsymbol{\pi},\boldsymbol{D}_0),\hat{t}_k}} \\
E_{(\boldsymbol{\pi},\boldsymbol{D}_0),\mathcal{T}^*}[N_{in+1}] &= \sum_{k=1}^{M} w_k \frac{\boldsymbol{t}(i)\boldsymbol{f}_{(\boldsymbol{\pi},\boldsymbol{D}_0),\hat{t}_k}(i)}{\boldsymbol{\pi}\boldsymbol{b}_{(\boldsymbol{\pi},\boldsymbol{D}_0),\hat{t}_k}}
\end{aligned} \tag{12}$$

The M-step needs not to be modified for fitting with an aggregated trace, i.e. Eq. 7 is used.

The steps for fitting PHDs with an EM algorithm and trace aggregation are summarized in Algorithm 1.

Algorithm 1 EM algorithm with trace aggregation for general PHDs

```
Input: Trace data $\mathcal{T} = t_1, \ldots, t_m$;
Output: PHD $(\boldsymbol{\pi}, \boldsymbol{D}_0)$;
Choose initial $(\boldsymbol{\pi}^{(0)}, \boldsymbol{D}_0^{(0)})$ and set $r = 0$;
Generate aggregated trace $\mathcal{T}^* = ((\hat{t}_1, w_1), (\hat{t}_2, w_2), \cdots (\hat{t}_M, w_M))$ from $\mathcal{T}$;
repeat
    Compute $\boldsymbol{f}_{(\boldsymbol{\pi}^{(r)}, \boldsymbol{D}_0^{(r)}), \hat{t}_i}$, $\boldsymbol{b}_{(\boldsymbol{\pi}^{(r)}, \boldsymbol{D}_0^{(r)}), \hat{t}_i}$ and $\boldsymbol{F}_{(\boldsymbol{\pi}^{(r)}, \boldsymbol{D}_0)^{(r)}, \hat{t}_i}$ for $i = 1, \cdots, M$ using Eq. 5;
    E-step: Compute the conditional expectations using Eq. 12;
    M-step: Compute $(\boldsymbol{\pi}^{(r+1)}, \boldsymbol{D}_0^{(r+1)})$ using Eq. 7;
    Set $r = r + 1$;
until $\|\boldsymbol{\pi}^{(r)} - \boldsymbol{\pi}^{(r-1)}\| + \|\boldsymbol{D}_0^{(r)} - \boldsymbol{D}_0^{(r-1)}\| < \epsilon$;
return $(\boldsymbol{\pi}^{(r)}, \boldsymbol{D}_0^{(r)})$;
```

3.2.2 Fitting MAPs:

In the following we extend Algorithm 1 to allow for MAP fitting with aggregated traces. Considering trace aggregation for MAPs is more sophisticated than for PHDs, because the forward and backward vectors are not computed independently for each trace element any more (cf. Eq. 9), but contain the complete joint density from the first to the i-th trace element (for the forward vectors) or the complete joint density from the i-th to the last trace element (for the backward vectors). Moreover, these vectors are also needed for the computation of the flow matrix.

Note from Eq. 10, that for the estimation of the M_{ij} only the forward and backward vectors are needed. The flow matrix is needed for estimating Z_i and N_{ij}. However, these values can also be estimated using Eq. 12 reusing values from the EM fitting with aggregation for PHDs.

For the EM approach for MAPs with aggregated traffic traces we need additional matrices $\boldsymbol{M}_{(\boldsymbol{\pi},\boldsymbol{D}_0),t} = e^{\boldsymbol{D}_0 t}, t = \hat{t}_1, \cdots, \hat{t}_M$ which can be be computed using uniformization as follows.

$$\boldsymbol{V}^{(0)} = \boldsymbol{I}, \quad \boldsymbol{V}^{(k+1)} = \boldsymbol{V}^{(k)}\boldsymbol{P}_0, \quad \boldsymbol{M}_{(\boldsymbol{\pi},\boldsymbol{D}_0),t} = \sum_{k=0}^{\infty} \beta(\alpha t, k)\boldsymbol{V}^{(k)}.$$

Then, we can define the forward and backward vectors for MAP fitting using the precomputed $\boldsymbol{M}_{(\boldsymbol{\pi},\boldsymbol{D}_0),t}$ and the sequence $\mathcal{S} = (s_1, s_2, \cdots, s_m)$.

$$\begin{aligned}
\tilde{\boldsymbol{ff}}^{(k)}_{(\boldsymbol{D}_0,\boldsymbol{D}_1)} &= \begin{cases} \boldsymbol{\pi} & \text{if } k = 1 \\ \tilde{\boldsymbol{ff}}^{(k-1)}_{(\boldsymbol{D}_0,\boldsymbol{D}_1)} \boldsymbol{M}_{(\boldsymbol{\pi},\boldsymbol{D}_0),\hat{t}_{s_{k-1}}} \boldsymbol{D}_1 & \text{if } 1 < k \leq m \end{cases} \\
\tilde{\boldsymbol{f}}^{(k)}_{(\boldsymbol{D}_0,\boldsymbol{D}_1)} &= \tilde{\boldsymbol{ff}}^{(k)}_{(\boldsymbol{D}_0,\boldsymbol{D}_1)} \boldsymbol{M}_{(\boldsymbol{\pi},\boldsymbol{D}_0),\hat{t}_{s_k}} \\
\tilde{\boldsymbol{bb}}^{(k)}_{(\boldsymbol{D}_0,\boldsymbol{D}_1)} &= \begin{cases} \boldsymbol{D}_1 \mathbb{1} & \text{if } k = m \\ \boldsymbol{D}_1 \boldsymbol{M}_{(\boldsymbol{\pi},\boldsymbol{D}_0),\hat{t}_{s_{k+1}}} \tilde{\boldsymbol{bb}}^{(k+1)}_{(\boldsymbol{D}_0,\boldsymbol{D}_1)} & \text{if } 1 \leq k < m \end{cases} \\
\tilde{\boldsymbol{b}}^{(k)}_{(\boldsymbol{D}_0,\boldsymbol{D}_1)} &= \boldsymbol{M}_{(\boldsymbol{\pi},\boldsymbol{D}_0),\hat{t}_{s_k}} \tilde{\boldsymbol{bb}}^{(k)}_{(\boldsymbol{D}_0,\boldsymbol{D}_1)}
\end{aligned} \tag{13}$$

Note, that the vectors are defined in a similar way as in Eqs. 8 and 9, but instead of computing the matrix exponential for the t_i we choose one of the precomputed $\boldsymbol{M}_{(\boldsymbol{\pi},\boldsymbol{D}_0),t}$ according to s_i which indicates the interval (and thus the $\hat{t}_j$) that t_i falls in.

The E-step uses these precomputed vectors and matrices:

$$\begin{aligned}
E_{(\boldsymbol{D}_0,\boldsymbol{D}_1),\mathcal{T}^*}[Z_i] &= \sum_{k=1}^{M} w_k \frac{\boldsymbol{F}_{(\boldsymbol{\pi},\boldsymbol{D}_0),\hat{t}_k}(i,i)}{\boldsymbol{\pi}\boldsymbol{b}_{(\boldsymbol{\pi},\boldsymbol{D}_0),\hat{t}_k}} \\
E_{(\boldsymbol{D}_0,\boldsymbol{D}_1),\mathcal{T}^*}[N_{ij}] &= \sum_{k=1}^{M} w_k \frac{\boldsymbol{D}_0(i,j)\boldsymbol{F}_{(\boldsymbol{\pi},\boldsymbol{D}_0),\hat{t}_k}(i,j)}{\boldsymbol{\pi}\boldsymbol{b}_{(\boldsymbol{\pi},\boldsymbol{D}_0),\hat{t}_k}} \qquad (14) \\
E_{(\boldsymbol{D}_0,\boldsymbol{D}_1),\mathcal{T}^*}[M_{ij}] &= \sum_{k=1}^{m-1} \frac{\tilde{\boldsymbol{f}}^{(k)}_{(\boldsymbol{D}_0,\boldsymbol{D}_1)}(i)\boldsymbol{D}_1(i,j)\tilde{\boldsymbol{b}}^{(k+1)}_{(\boldsymbol{D}_0,\boldsymbol{D}_1)}(j)}{\boldsymbol{\pi}\tilde{\boldsymbol{b}}^{(1)}_{(\boldsymbol{D}_0,\boldsymbol{D}_1)}} + \frac{\tilde{\boldsymbol{f}}^{(m)}_{(\boldsymbol{D}_0,\boldsymbol{D}_1)}(i)\boldsymbol{D}_1(i,j)\mathbb{1}}{\boldsymbol{\pi}\tilde{\boldsymbol{b}}^{(1)}_{(\boldsymbol{D}_0,\boldsymbol{D}_1)}}
\end{aligned}$$

And finally, the M-step becomes

$$\begin{aligned}
\hat{\boldsymbol{D}}_0(i,j) &= \frac{E_{(\boldsymbol{D}_0,\boldsymbol{D}_1),\mathcal{T}^*}[N_{ij}]}{E_{(\boldsymbol{D}_0,\boldsymbol{D}_1),\mathcal{T}^*}[Z_i]} \quad \text{for } i \neq j, \;\; \hat{\boldsymbol{D}}_1(i,j) = \frac{E_{(\boldsymbol{D}_0,\boldsymbol{D}_1),\mathcal{T}^*}[M_{ij}]}{E_{(\boldsymbol{D}_0,\boldsymbol{D}_1),\mathcal{T}^*}[Z_i]} \\
\hat{\boldsymbol{D}}_0(i,i) &= -\left(\sum_{j=1,i\neq j}^{n} \hat{\boldsymbol{D}}_0(i,j) + \sum_{j=1}^{n} \hat{\boldsymbol{D}}_1(i,j)\right) \qquad (15)
\end{aligned}$$

Vector $\boldsymbol{\pi}$ is assumed to be the embedded stationary distribution after an event such that $\hat{\boldsymbol{\pi}}$ is the solution of $\hat{\boldsymbol{\pi}}\left(-\hat{\boldsymbol{D}}_0\right)^{-1}\hat{\boldsymbol{D}}_1 = \hat{\boldsymbol{\pi}}$ subject to $\hat{\boldsymbol{\pi}}\mathbb{1} = 1$.

Algorithm 2 summarizes our approach.

Algorithm 2 EM algorithm with trace aggregation for MAPs

```
Input:  Trace data $\mathcal{T} = t_1, \ldots, t_m$;
Output:  MAP $(\boldsymbol{D}_0, \boldsymbol{D}_1)$;
Choose initial $(\boldsymbol{D}_0^{(0)}, \boldsymbol{D}_1^{(0)})$ and set $r = 0$;
Generate aggregated trace $\mathcal{T}^* = ((\hat{t}_1, w_1), (\hat{t}_2, w_2), \cdots (\hat{t}_M, w_M))$ and $\mathcal{S} = (s_1, s_2, \cdots, s_m)$ from $\mathcal{T}$;
repeat
    Compute $\boldsymbol{f}_{(\boldsymbol{\pi}^{(r)},\boldsymbol{D}_0^{(r)}),\hat{t}_i}$, $\boldsymbol{b}_{(\boldsymbol{\pi}^{(r)},\boldsymbol{D}_0^{(r)}),\hat{t}_i}$, $\boldsymbol{F}_{(\boldsymbol{\pi}^{(r)},\boldsymbol{D}_0)^{(r)},\hat{t}_i}$ and $\boldsymbol{M}_{(\boldsymbol{\pi},\boldsymbol{D}_0),\hat{t}_i}$ for $i = 1, \cdots, M$;
    Compute $\tilde{\boldsymbol{f}}^{(k)}_{(\boldsymbol{D}_0,\boldsymbol{D}_1)}$ and $\tilde{\boldsymbol{b}}^{(k)}_{(\boldsymbol{D}_0,\boldsymbol{D}_1)}$ for $k = 1, \cdots, m$ according to Eq. 13;
    E-step: Compute the conditional expectations using Eq. 14;
    M-step: Compute $(\boldsymbol{D}_0^{(r+1)}, \boldsymbol{D}_1^{(r+1)})$ using Eq. 15;
    Set $r = r + 1$;
until $\|\boldsymbol{D}_0^{(r)} - \boldsymbol{D}_0^{(r-1)}\| + \|\boldsymbol{D}_1^{(r)} - \boldsymbol{D}_1^{(r-1)}\| < \epsilon$;
return $(\boldsymbol{D}_0^{(r)}, \boldsymbol{D}_1^{(r)})$;
```

For the EM approach without trace aggregation the most time consuming part is the computation of the m forward vectors, the m backward vectors and the m flow matrices using e.g. randomization. For the approach with trace aggregation we have to compute M flow matrices with $M << m$ using randomization. Additionally, M matrix exponentials have to be computed and stored. Since $\boldsymbol{f}_{(\boldsymbol{\pi},\boldsymbol{D}_0),t} = \boldsymbol{\pi}\boldsymbol{M}_{(\boldsymbol{\pi},\boldsymbol{D}_0),t}$ and $\boldsymbol{b}_{(\boldsymbol{\pi},\boldsymbol{D}_0),t} = \boldsymbol{M}_{(\boldsymbol{\pi},\boldsymbol{D}_0),t}\boldsymbol{t}$ computation of those forward and backward vectors is cheap once $\boldsymbol{M}_{(\boldsymbol{\pi},\boldsymbol{D}_0),t}$ is known. The computation of $\tilde{\boldsymbol{f}}^{(k)}_{(\boldsymbol{D}_0,\boldsymbol{D}_1)}$ and $\tilde{\boldsymbol{b}}^{(k)}_{(\boldsymbol{D}_0,\boldsymbol{D}_1)}$ only consists of vector-matrix multiplications which is relatively cheap compared to the computation of the matrix exponential.

4 Experimental Results

For assessing the fitting quality of the presented EM algorithm we performed several experiments with synthetically generated traces and with traces from real systems. The synthetically generated traces have been created from a MAP and should be easier to fit than the real traces. However, the results provide hints on how many intervals are necessary for the algorithm to obtain good results.

For traces from computer and communication networks, [26] recommends using logarithmic trace aggregation, because with uniform trace aggregation few intervals contain the major part of the trace elements and many intervals contain only little or no trace elements. Consequently, we only present results using logarithmic trace aggregation. All experiments have been performed on a dual-core AMD Opteron Processor with 2.4 GHz and 16 GB RAM.

4.1 Synthetically Generated Traces

We generated 500.000 observations from the following MAP

$$\boldsymbol{D}_0 = \begin{bmatrix} -1.733 & 0.064 & 0.923 \\ 0.159 & -2.289 & 0.040 \\ 2.015 & 0.101 & -2.344 \end{bmatrix}, \qquad \boldsymbol{D}_1 = \begin{bmatrix} 0.746 & 0.0 & 0.0 \\ 0.0 & 2.090 & 0.0 \\ 0.0 & 0.0 & 0.228 \end{bmatrix}$$

and fitted MAPs of order 3 using different numbers of intervals between 100 and 2000. Table 1 shows the log-likelihood values for the different interval numbers after 100 iterations of the EM algorithm. The results show that the best value was reached with 1000 intervals and cannot be further improved when increasing the number of intervals. For comparison, the original $MAP(3)$ has a log-likelihood value of -460207 for the trace. Using 1000 intervals our algorithm returned the following MAP

Intervals	100	500	1000	2000
Log-likelihood	-460897	-460860	-460859	-460859

Table 1: Log-likelihood values for different interval numbers

$$\boldsymbol{D}_0 = \begin{bmatrix} -2.4646 & 0.222 & 0.051 \\ 0.449 & -1.4654 & 0.639 \\ 0.051 & 0.126 & -0.75706 \end{bmatrix}, \qquad \boldsymbol{D}_1 = \begin{bmatrix} 2.059 & 0.132 & 0.0006 \\ 0.145 & 0.222 & 0.0104 \\ 0.00006 & 0.004 & 0.576 \end{bmatrix}$$

Fig. 1 shows the log-likelihood values (left y-axis) and the total time consumption (right y-axis) after each of the 100 iterations. Note, that the log-likelihood values are the values for the original trace and not for the aggregated trace. As one can see, the largest improvement in the likelihood was obtained during the first 10 iterations. The time consumption is almost linear and overall the algorithm required about 40 seconds. Fig. 2 shows the probability density function and the lag-k autocorrelation coefficients for the original $MAP(3)$, the generated trace and the fitted MAP. As one can see, the density functions almost overlap completely, however, the fitted MAP slightly underestimated the autocorrelation.

4.2 Traces from a Real System

As real traces we used the two standard benchmark traces BC-pAug89 [18] and LBL-TCP-3 [27] from the Internet traffic archive (http://ita.ee.lbl.gov). The traces contain $1.000.000$ and about $1.700.000$ entries, respectively, and fitting them using

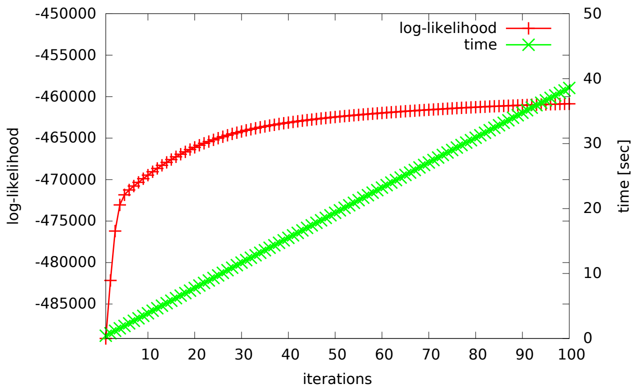
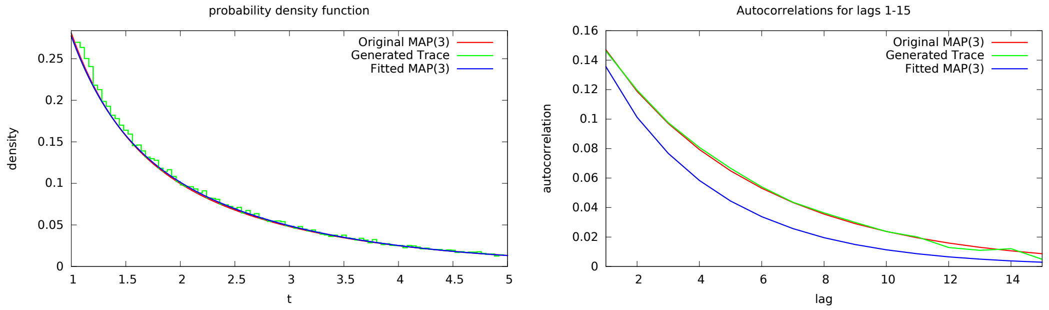


Figure 1: Log-likelihood and time consumption for fitting a synthetically generated trace with a $MAP(3)$ and 1000 intervals



(a) probability density function

(b) lag-k autocorrelation

Figure 2: Density and autocorrelations for the original MAP, the generated trace and the fitted MAP

an EM algorithm is very time consuming because of their size. We fitted MAPs of different order with a different number of intervals using our approach with trace aggregation. For comparison we also fitted MAPs using the EM algorithm from [4] that does not use trace aggregation and served as basis for our implementation. Fig. 3 shows the results for MAPs of order 5 fitted to the trace LBL-TCP-3. On the left y-axis the log-likelihood values are displayed, the right y-axis shows the total amount of time used for 50 iterations of the algorithms. The curves labeled with `agg` result from MAP EM fitting with trace aggregation using 1000 intervals, the curves labeled with `no agg` show the corresponding results from EM fitting without trace aggregation. As one can see, the log-likelihood values are almost identical for both algorithms, however, without trace aggregation the algorithm needed over 8 hours and with trace aggregation the 50 iterations have been computed in less than 10 minutes.

To introduce another measure for comparing the fitting results aside from the likelihood we used the trace and the two fitted MAPs as arrival processes in a simple single-server queueing system with a capacity of 10 and exponentially distributed service times. Fig. 4 shows the queue length distribution from simulation runs with a utilization of $\rho = 0.5$ and $\rho = 0.8$. In both cases the two MAPs slightly overestimate the middle part of the queue length distribution resulting from the trace, but as expected from the almost identical likelihood values the two MAPs behave similar.

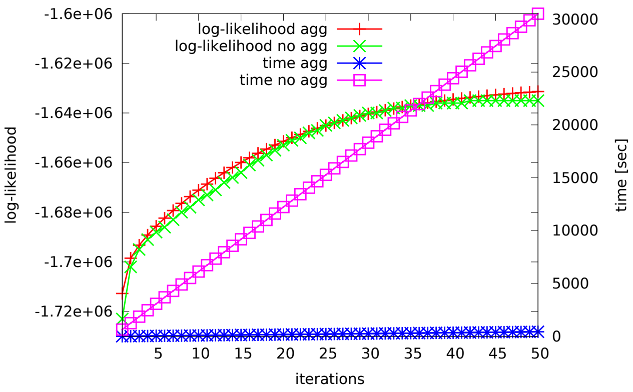
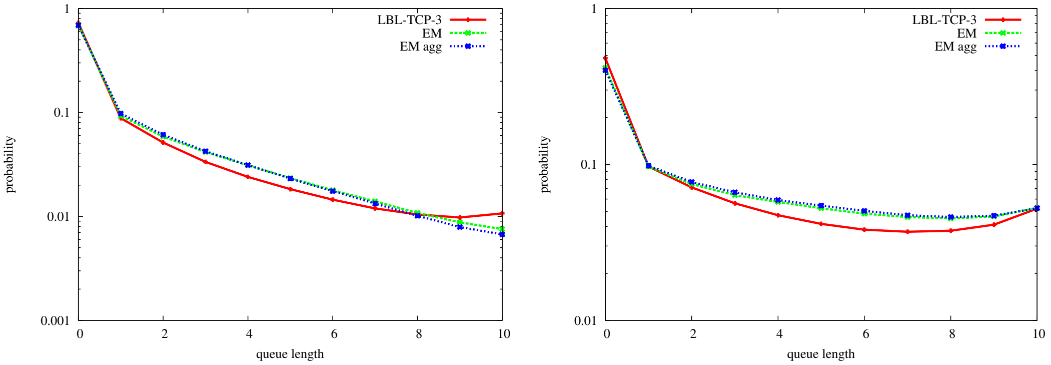


Figure 3: Log-likelihood and time consumption for the trace LBL-TCP-3 and EM fitting with and without trace aggregation



(a) $\rho = 0.5$

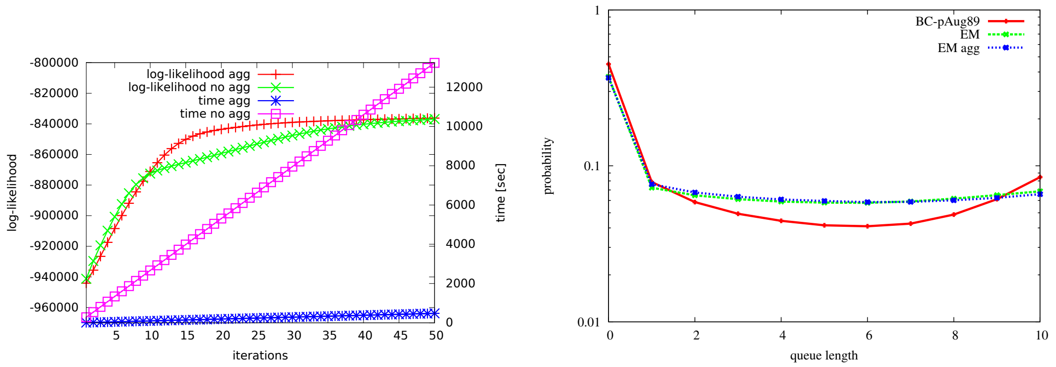
(b) $\rho = 0.8$

Figure 4: Queueing Results for the trace LBL-TCP-3 and two fitted MAPs

Fig. 5 shows the result for the trace BC-pAug89. Fig. 5(a) compares the time consumption and the log-likelihood when fitting MAPs of order 6 using an EM algorithm with and without trace aggregation, respectively, for the trace, which gives a similar picture as the results for the trace LBL-TCP-3. Fig. 5(b) contains the queueing results. Again both MAPs behave similarly.

5 Conclusions

The paper presents an EM algorithm for parameter fitting of *Phase Type Distributions* (PHDs) or *Markovian Arrival Processes* (MAPs). In contrast to known approaches, elements in the trace are first aggregated and then the EM algorithm is applied. A similar approach has been proposed before for PHDs but not for MAPs. Our experiments show that trace aggregation results in a dramatical improvement of the fitting times when the traces contain more than a few thousand elements. The value of the likelihood is not affected by trace aggregation which implies that the quality of the generated PHD or MAP is the same as if the process results from an EM algorithm applied to the original trace.



(a) Log-likelihood and time consumption

(b) Queueing results for $\rho = 0.8$

Figure 5: Results for the trace BC-pAug89

The approach can be extended by considering only specific matrix structures like acyclic matrices $\boldsymbol{D}_0$ or use the EM algorithm for PHDs to determine the parameters of an acyclic distribution in canonical form as done in [24]. Furthermore the approach can also be extended to fit the parameters of PHDs or MAPs according to grouped data as in [23, 25].

References

[1] J. R. Artalejo, A. Gomez-Corral, and Q. M. He. Markovian arrivals in stochastic modelling: a survey and some new results. *Sort: Statistics and Operations Research Transactions*, 34(2):101–156, 2010.

[2] S. Asmussen, O. Nerman, and M. Olsson. Fitting phase-type distributions via the EM-algorithm. *Scand. J. Stat.*, 23(4):419–441, 1996.

[3] L. Breuer. An EM algorithm for batch Markovian arrival processes and its comparison to a simpler estimation procedure. *Annals OR*, 112(1-4):123–138, 2002.

[4] P. Buchholz. An EM-algorithm for MAP fitting from real traffic data. In *Computer Performance Evaluation / TOOLS*, pages 218–236. Springer, 2003.

[5] P. Buchholz, I. Felko, and J. Kriege. *Input Modeling with Phase Type Distributions and Markov Models*. Springer, to appear.

[6] P. Buchholz and J. Kriege. A heuristic approach for fitting MAPs to moments and joint moments. In *QEST*, pages 53–62, 2009.

[7] W. Bux and U. Herzog. The phase concept: approximation of measured data and performance analysis. In *Computer Performance*, pages 23–38, 1977.

[8] G. Casale, E. Z. Zhang, and E. Smirni. KPC-toolbox: Simple yet effective trace fitting using Markovian arrival processes. In *QEST*, pages 83–92, 2008.

[9] A. Feldmann and W. Whitt. Fitting mixtures of exponentials to long-tail distributions to analyze network performance models. *Performance Evaluation*, 1998.

[10] I. Gerhardt and B. L. Nelson. On capturing dependence in point processes: Matching moments and other techniques. Technical report, Northwestern Univ., 2009.

[11] A. Heindl and M. Telek. Output models of MAP/PH/1(/K) queues for an efficient network decomposition. *Perform. Eval.*, 49(1/4):321–339, 2002.

[12] A. Horváth and M. Telek. Matching more than three moments with acyclic phase type distributions. *Stochastic Models*, 23:167–194, 2007.

[13] G. Horváth, M. Telek, and P. Buchholz. A MAP fitting approach with independent approximation of the inter-arrival time distribution and the lag-correlation. In *QEST*, pages 124–133. IEEE CS Press, 2005.

[14] M.A. Johnson and M.R. Taaffe. Matching moments to phase distributions: Nonlinear programming approaches. *Stochastic Models*, 2(6):259–281, 1990.

[15] R. El Abdouni Khayari, R. Sadre, and B. Haverkort. Fitting world-wide web request traces with the EM-algorithm. *Performance Evaluation*, 52:175–191, 2003.

[16] A. Klemm, C. Lindemann, and M. Lohmann. Modeling IP traffic using the batch Markovian arrival process. *Perform. Eval.*, 54(2):149–173, 2003.

[17] J. Kriege and P. Buchholz. An Empirical Comparison of MAP Fitting Algorithms. In *Proc. of the 15th MMB & DFT*, 2010.

[18] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson. On the self-similar nature of ethernet traffic (extended version). *IEEE/ACM Trans. Netw.*, 1994.

[19] G.J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. John Wiley and Sons, 1997.

[20] M. F. Neuts. A versatile Markovian point process. *Jour. of Appl. Probability*, 1979.

[21] M. F. Neuts. *Matrix-geometric solutions in stochastic models*. Johns Hopkins University Press, 1981.

[22] C. A. O'Cinneide. Phase-type distributions: open problems and a few properties. *Stochastic Models*, 15(4):731–757, 1999.

[23] H. Okamura, T. Dohi, and K.S. Trivedi. Markovian arrival process parameter estimation with group data. *IEEE/ACM Trans. Netw.*, 17(4):1326–1339, 2009.

[24] H. Okamura, T. Dohi, and K.S. Trivedi. A refined EM algorithm for PH distributions. *Perform. Eval.*, 68(10):938–954, 2011.

[25] H. Okamura, T. Dohi, and K.S. Trivedi. Improvement of expectation-maximization algorithm for phase-type distributions with grouped and truncated data. *Applied Stochastic Models in Business and Industry*, 29(2):141–156, 2012.

[26] A. Panchenko and A. Thümmler. Efficient phase-type fitting with aggregated traffic traces. *Perform. Eval.*, 64(7-8):629–645, 2007.

[27] V. Paxson and S. Floyd. Wide area traffic: the failure of Poisson modeling. *IEEE/ACM Trans. Netw.*, 3(3):226–244, 1995.

[28] W. J. Stewart. *Introduction to the numerical solution of Markov chains*. Princeton University Press, 1994.

[29] A. Thümmler, P. Buchholz, and M. Telek. A novel approach for phase-type fitting with the EM algorithm. *IEEE Trans. Dep. Sec. Comput.*, 3(3):245–258, 2006.