A Novel Approach for Fitting Probability Distributions to Real Trace Data with the EM Algorithm

Axel Thümmler and Peter Buchholz
University of Dortmund
Department of Computer Science
August-Schmidt-Str. 12
44227 Dortmund, Germany
{thuemmler, buchholz}@ls4.cs.uni-dortmund.de

Miklós Telek
Budapest University of Tech. and Econ.
Department of Telecommunications
Magyar Tudósok krt. 2
1117 Budapest, Hungary
telek@hit.bme.hu

Abstract

The representation of general distributions or measured data by phase-type distributions is an important and non-trivial task in analytical modeling. Although a large number of different methods for fitting parameters of phase-type distributions to data traces exist, many approaches lack efficiency and numerical stability. In this paper, a novel approach is presented that fits a restricted class of phase-type distributions, namely mixtures of Erlang distributions, to trace data. For the parameter fitting an algorithm of the expectation maximization type is developed. The paper shows that these choices result in a very efficient and numerically stable approach which yields phase-type approximations for a wide range of data traces that are as good or better than approximations computed with other less efficient and less stable fitting methods. To illustrate the effectiveness of the proposed fitting algorithm, we present comparative results for our approach and two other methods using six benchmark traces and two real traffic traces.

1. Introduction

The central idea of traffic modeling lies in constructing analytically tractable models that capture the most important statistical properties of an underlying measured data trace. For analytical performance and reliability modeling measured data has to be represented or approximated by phase-type (PH) distributions in several cases. The procedure of computing or estimating the parameters of a phase-type distribution according to some sample data or with respect to some other known distribution is commonly denoted as phase-type fitting.

Among the large number of available fitting methods, expectation-maximization (EM) algorithms [15] are general methods of finding the maximum-likelihood estimate of the parameters of an underlying distribution from a given data trace when the data is incomplete or has missing values. EM algorithms for phase-type fitting are available for some time [1], [4] but the application of the basic approach to general PH distributions turns out to be extremely costly and the fitted distribution depends heavily on the initial values [16]. Thus, it seems that fitting general PH distributions is not appropriate if the number of phases increases above 4, which is often the case for small coefficients of variation or traces that cannot be adequately represented by a PH distribution of low order. To overcome these problems the class of PH distributions used for fitting has to be restricted which is in principle possible in the basic EM algorithm by initializing only some elements in the matrix with non-zero values, but it seems to be more appropriate to develop an EM algorithm tailored to specific types of PH distributions. Based on earlier work from [9], El Abdouni Khayari et. al. developed an EM algorithm in [7] to fit the parameters of an hyperexponential distribution to values of a data trace. The resulting approach is extremely efficient and yields good fitting results for heavy tailed distributions with monotonically decreasing density functions. However, the use of hyperexponential distributions restricts the class of distributions, which can be represented. In fact, hyperexponential distributions cannot adequately capture general distributions with increasing and decreasing densities or with a coefficient of variation less than one.

Since the fitting of parameters of a PH distribution is in general a non-linear optimization problem, apart from the EM algorithm also other optimization

algorithms can be applied. However, the optimization problem for general PH distributions is too complex to yield satisfactory results, if the number of phases is larger than two or three. As shown in several papers [2], [3], [11], [12], the fitting problem becomes much easier if acyclic instead of general phase-type distributions are used, because for this type of distributions a canonical representation exists which reduces the number of free parameters to 2N compared to $2N^2$ for the general case, where N is then number of phases [5]. On the other hand, the restriction to acyclic PH distributions seems not to limit the flexibility of the approach. However, even in the acyclic case, the resulting optimization is still complex and contains local optima and saddle points. To overcome the problem of convergence to a local optimum, the fitting algorithm is usually started with several initial settings and the best fitting is chosen.

Apart from acyclic phase-type distributions several other restricted classes have been used. For our approach the work of Johnson [14] and Schmickler [19] are most important, since both use mixtures of Erlang distributions, which are also used in our work and will be denoted as hyper-Erlang distributions (HErD) according to [8]. However, in contrast to our approach, the mentioned techniques fit some moments and specific properties of the distribution or density function using nonlinear optimization.

In this paper, an EM algorithm for the fitting of hyper-Erlang distributions is presented. The approach, which will be denoted as G-FIT, extends the fitting procedure of [7] from hyperexponential to hyper-Erlang distributions, which extend the class of representable distributions significantly since mixtures of Erlang distributions of unlimited order are theoretically as powerful as acyclic or general PH distributions (see Theorem 1). However, the class of distributions still allows the realization of a very efficient fitting algorithm. In particular the fitting time is independent of the number of states, it depends only on the number of Erlang branches, which might be significantly lower than the number of states. In fact, for M Erlang branches and a trace with K samples the time complexity of our algorithm is in O(M·K). Thus, distributions with a large number of states can be fitted efficiently. Furthermore, the fitting algorithm is rather stable due to the specific structure of the density function, which yields a fast and reliable convergence of the EM method.

Apart from the efficiency of the approach, the quality of the approximation for a given number of phases is important. We tested the approach on a set of benchmark traces [3] and compared it with general PH-fitting [1] and fitting of acyclic PH distributions [12]. As expected G-FIT is significantly faster than the other two approaches. Additionally, we were able to reach with an identical number of states a similar or better fitting quality than with the other two approaches on almost all examples. This result was not expected, because hyper-Erlang distributions of a given order are in general less flexible than acyclic or general PH distributions of the same order. The practical applicability of G-FIT is demonstrated by fitting a call center trace [17] and a large TCP traffic trace [13] with more than 10^6 samples. The presented EM algorithm is implemented in the software package G-FIT, which is available for download on the Web [10].

The paper is organized as follows. Section 2 introduces the class of hyper-Erlang distributions. Section 3 develops a specialized EM algorithm for fitting the continuous parameters of a hyper-Erlang distribution and Section 4 presents an approach for finding optimal settings of the discrete parameters of the distribution. Experimental results obtained from fitting six synthetically generated benchmark traces and real traffic traces are presented in Section 5. Finally, concluding remarks are given.

2. Hyper-Erlang Distributions

We consider a mixture of M mutually independent Erlang distributions weighted with the (initial) probabilities $\alpha_1, \ldots, \alpha_M$ with $0 < \alpha_m \leq 1$ and $\alpha_1+\alpha_2+\ldots+\alpha_M=1$. The *number of phases* of the m-th Erlang distribution is denoted with r_m. We assume $r_1 \leq \ldots \leq r_M$ without loss of generality. Furthermore, let λ_m be the *scale parameter* of the m-th Erlang distribution. Note, that the individual Erlang distributions need not have the same mean. According to [8], we call this mixture of Erlang distributions a *hyper-Erlang distribution (HErD)*. The HErD belongs to the class of acyclic phase-type distributions [2]. Besides the Erlang distribution, for M = 1, the hyperexponential distribution is a special case of a HErD with $r_m = 1$ for all m=1,…,M.

Let X be a hyper-Erlang random variable. The probability density function (pdf) for X is given by

$$f_X(x) = \sum_{m=1}^{M} \alpha_m \frac{(\lambda_m x)^{r_m-1}}{(r_m-1)!} \lambda_m e^{-\lambda_m x}, \qquad (1)$$

and the i-th moment $E[X^i]$ is given by

$$E[X^i] = \sum_{m=1}^{M} \alpha_m \frac{(r_m+i-1)!}{(r_m-1)!} \frac{1}{\lambda_m^i}. \qquad (2)$$

A common measure to characterize the flexibility in approximating a given general distribution function is

the range of variability of the squared coefficient of variation, which is defined by $c_X^2 = E[X^2]/E[X]^2 - 1$.

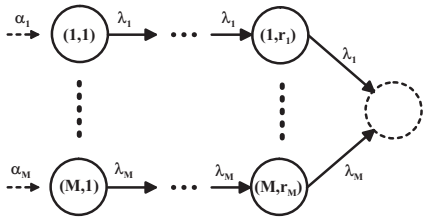

Fig. 1. State transition graph of a HErD

Fig. 1 shows the state transition graph of a HErD, which corresponds to an absorbing continuous-time Markov chain where a state change occurs after an exponentially distributed delay with mean $1/\lambda_m$, m=1,…,M, and the time until absorption has a HErD. The absorbing state is shown as a dashed circle in Fig. 1. The *number of states* of a HErD is the overall number of exponential distributions involved in its construction, which is given by $N = \sum_{m=1}^{M} r_m$.

Let f(x; M, **r**, **α**, **λ**) denote the density function of the HErD with M Erlang branches, where $\mathbf{r} = (r_1, r_2, \ldots, r_M) \in \mathbb{N}^M$ is a vector containing the number of phases of each Erlang branch, $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_M) \in \mathbb{R}^M$ is a vector with the initial probabilities for each Erlang branch and $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_M) \in \mathbb{R}^M$ is a vector with the scaling parameters respectively. Furthermore, let $\mathcal{H}_N$ be a set of all HErD having N states, i.e.,

$$\mathcal{H}_N = \left\{ f(x; M, \mathbf{r}, \boldsymbol{\alpha}, \boldsymbol{\lambda}) \middle| 1 \le M \le N,\ \lambda_m > 0,\right.$$
$$\left. \alpha_m \ge 0,\ r_m \ge 1,\ \sum_{m=1}^{M} \alpha_m = 1,\ \sum_{m=1}^{M} r_m = N \right\}. \quad (3)$$

In fact, the set $\mathcal{H}_N$ contains all HErD distributions having at most N states, since HErD with less than N states are obtained by simply setting some α_m values to zero. The versatility of the HErD in approximating general distributions is shown by the following theorem.

Theorem 1:

(i) Let $\mathcal{F}$ denote the set of all probability density functions of nonnegative random variables, then $\mathcal{H}_\infty$ is a dense set in $\mathcal{F}$, i.e., for every density function $f \in \mathcal{F}$ it is possible to choose a sequence of density functions $f_n(x) \in \mathcal{H}_n$, such that $\lim_{n\to\infty} f_n(x) = f(x)$ for all x at which f(x) is continuous.

(ii) Let f be a hyper-Erlang distribution out of the set $\mathcal{H}_N$, with $N \ge 2$. The parameters of f can be tuned such that the squared coefficient of variation of f equals 1/N or takes on an arbitrary value greater or equal to 1/(N-1) with f still being an element of $\mathcal{H}_N$.

Proof: The proof of (i) can be found in [8]. In particular, the construction of a general probability density function from an infinite mixture of Erlang densities is based on appropriately choosing the weights α_m. For the proof of (ii) we refer the reader to the extended version of this paper, which is available for download on the Web [10]. ■

Note that Theorem 1 states that any probability density function of a nonnegative random variable can be approximated by a hyper-Erlang distribution. Next we intend to shed some light onto the relationship between sub-classes of PH distributions. Let $\mathcal{A}$ and $\mathcal{B}$ sets of specific PH distributions. We consider three types of relationships between sets $\mathcal{A}$ and $\mathcal{B}$.

(i) $\mathcal{A} < \mathcal{B}$ means that all finite-state distributions of $\mathcal{A}$ can be represented by an appropriately selected finite-state distribution of $\mathcal{B}$ and $\mathcal{B}$ contains at least one distribution that cannot be represented by a distribution of $\mathcal{A}$ even with an infinite number of states.

(ii) $\mathcal{A} \le_\infty \mathcal{B}$ means that all finite-state distributions of $\mathcal{A}$ can be represented by an appropriately selected finite-state distribution of $\mathcal{B}$ and $\mathcal{B}$ contains at least one distribution that can only be represented by a distribution of $\mathcal{A}$ with an infinite number of states.

(iii) $\mathcal{A} \ne \mathcal{B}$ means that none of the relationships (i) and (ii) hold.

Note, that relationship (i) means that a distribution of $\mathcal{B}$ cannot be approximated arbitrarily close by a distribution of $\mathcal{A}$, whereas in relationship (ii) this approximation is possible. According to this definition, we consider the relationship between some well-known sub-classes of phase-type distributions and their versatility in representing general distributions. In particular we consider exponential distributions (ED), hyperexponential distributions (HED), Erlang distributions (ErD), hyper-Erlang distributions (HErD), hypoexponential distributions (HoED), acyclic phase-type distributions (APHD), and phase-type distributions (PHD). A detailed definition of these distributions as well as the computation of their squared coefficient of variation can be found in standard textbooks (see e.g. [20]).

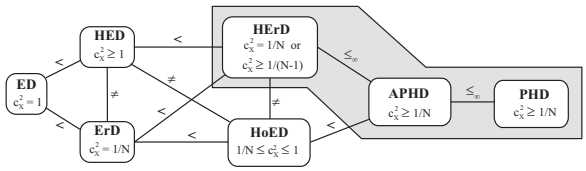

Fig. 2. Relationship of sub-classes of PH

Fig. 2 shows the relationship between the distributions introduced above according to the relationships (i) – (iii). The classes of distributions that are dense in the set of general distributions, i.e., HErD, APHD, and PHD, are combined in the gray shaded area. The relationships can be explained by comparing the possible range of the squared coefficient of variation that a distribution can take on. We conclude from this comparison that HErD is the most versatile sub-class of APHD, since HErD also provides full flexibility but can be more efficiently tuned to match general distributions than APHD as shown in the next sections.

3. Fitting Hyper-Erlang Distributions

3.1 The EM Algorithm for Mixture-Densities

The mixture-density parameter estimation problem is probably one of the most widely used applications of the EM algorithm [6]. In this case, we assume the following probabilistic model

$$p(x|\Theta) = \sum_{m=1}^{M} \alpha_m p_m(x|\theta_m), \quad (4)$$

where the parameters are $\Theta = (\alpha_1,\ldots,\alpha_M, \theta_1,\ldots,\theta_M)$ such that $\alpha_1+\alpha_2+\ldots+\alpha_M=1$ and each p_m is a density function parameterized by θ_m. In other words, we assume that M component densities are mixed together with M mixing coefficients α_m. Note that in general θ_m can be a vector of parameters for each density function p_m, but it is a single value in our HErD fitting method.

Let $\mathcal{T} = \{x_1,\ldots,x_K\}$ be a data set of measurements supposedly drawn from the distribution (4). That is, we assume that these data values are drawn from independent and identically distributed random variables with probability density function (4). The log-likelihood expression for this mixture density for the trace $\mathcal{T}$ is given by

$$\log L(\Theta|\mathcal{T}) = \log \prod_{k=1}^{K} p(x_k|\Theta) = \sum_{k=1}^{K} \log\left(\sum_{m=1}^{M} \alpha_m p_m(x_k|\theta_m)\right), \quad (5)$$

which is difficult to optimize because it contains the logarithm of a sum. If we consider $\mathcal{T}$ as incomplete data and assume the existence of unobserved data items $y_k \in \{1,\ldots,M\}$, k=1,…,K, whose values inform us which component density "generates" each data item of $\mathcal{T}$, the likelihood expression can be significantly simplified. That is, we assume $y_k = m$ if the k-th sample x_k was generated by the m-th mixture component p_m. If we know the values $\mathbf{y} = (y_1,\ldots,y_K)$ the log-likelihood expression of Eq. (5) becomes

$$\log L(\Theta|\mathcal{T},\mathbf{y}) = \sum_{k=1}^{K} \log\left(\alpha_{y_k} p_{y_k}(x_k|\theta_{y_k})\right). \quad (6)$$

The problem in dealing with Eq. (6) is, that we do not know the values of y_k. If we assume y_k as random values drawn from a random variable Y, we can derive an expression for the probability mass function (pmf), denoted by q(y), of the unobserved data. First, we guess at parameters for the mixture density, i.e., we guess that $\hat{\Theta} = (\hat{\alpha}_1,\ldots,\hat{\alpha}_M,\hat{\theta}_1,\ldots,\hat{\theta}_M)$ are the appropriate parameters. Given $\hat{\Theta}$, we can easily compute the mixture components $p_m(x_k|\hat{\theta}_m)$ for each k and m. Keeping in mind that α_m is the probability of choosing the m-th mixture component we can compute the pmf of the unobserved data given the observed data $\mathcal{T}$ and the estimates $\hat{\Theta}$ using Bayes's rule

$$q(y_k|x_k,\hat{\Theta}) = q(y_k|\hat{\Theta})\cdot p(x_k|y_k,\hat{\Theta}) \Big/ p(x_k|\hat{\Theta}) = \hat{\alpha}_{y_k}\cdot p_{y_k}(x_k|\hat{\theta}_{y_k}) \Big/ \sum_{m=1}^{M} \hat{\alpha}_m \cdot p_m(x_k|\hat{\theta}_m), \quad (7)$$

and

$$q(\mathbf{y}|\mathcal{T},\hat{\Theta}) = \prod_{k=1}^{K} q(y_k|x_k,\hat{\Theta}), \quad (8)$$

where $\mathbf{y} \in \{1,\ldots,M\}^K$ is an instance of the unobserved data independently drawn from Y. The expected value of the complete-data log-likelihood with respect to the unknown random variable Y given the observed data $\mathcal{T}$ and the current parameter estimates $\hat{\Theta}$, is given by

$$Q(\Theta,\hat{\Theta}) = E\left[\log L(\Theta|\mathcal{T},Y)\,\middle|\,\mathcal{T},\hat{\Theta}\right] = \sum_{\mathbf{y}\in\{1,\ldots,M\}^K} \log L(\Theta|\mathcal{T},\mathbf{y})\cdot q(\mathbf{y}|\mathcal{T},\hat{\Theta}). \quad (9)$$

Inserting Eqs. (6) and (8) into Eq. (9) we get

$$Q(\Theta,\hat{\Theta}) = \sum_{\mathbf{y}\in\{1,\ldots,M\}^K} \sum_{k=1}^{K} \log\left(\alpha_{y_k} p_{y_k}(x_k|\theta_{y_k})\right) \cdot \prod_{k=1}^{K} q(y_k|x_k,\hat{\Theta}), \quad (10)$$

and rearranging the sums and the product results in

$$Q(\Theta,\hat{\Theta}) = \sum_{m=1}^{M}\sum_{k=1}^{K} \log(\alpha_m)\cdot q(m|x_k,\hat{\Theta}) + \sum_{m=1}^{M}\sum_{k=1}^{K} \log\left(p_m(x_k|\theta_m)\right)\cdot q(m|x_k,\hat{\Theta}). \quad (11)$$

Note that the computation of the expectation in Eq. (9) constitutes the E-step of the EM algorithm. In general, the main difficulty in computing this expectation is to obtain an expression for the marginal distribution of the unobserved data. However, for the mixture density problem discussed in this section the marginal distribution can be simply computed by Eqs. (7) and (8). The M-step of the EM algorithm is to

maximize the expectation computed in the E-step with respect to Θ. To maximize Eq. (11), we can maximize the term containing α_m (first sum in Eq. (11)) and the term containing θ_m (second sum in Eq. (11)) independently since they are not related. According to [15], a Lagrange multiplier can be applied to find the expression for α_m, resulting in

$$\alpha_m = \frac{1}{K}\sum_{k=1}^{K} q\left(m\middle|x_k,\hat{\Theta}\right). \qquad (12)$$

The computation of θ_m depends on the form of the mixture density component p_m and is addressed in the next section for a mixture of Erlang distributions.

3.2 Application to Hyper-Erlang Distributions

In this section we develop the formulas for application of the EM algorithm to the mixture density parameter estimation problem when a mixture component is an Erlang distribution, i.e.,

$$p_m\left(x_k\middle|\lambda_m\right) = \frac{(\lambda_m x_k)^{r_m-1}}{(r_m-1)!}\lambda_m e^{-\lambda_m x_k}, \qquad (13)$$

and the mixture distribution is described by the parameter vector $\Theta = (\alpha_1,\ldots,\alpha_M,\lambda_1,\ldots,\lambda_M)$. The parameters α_m, m=1,…,M, that maximize Eq. (11) are determined according to Eq. (12). In order to determine the parameters λ_m, m=1,…,M, that maximize Eq. (11) we set the derivatives with respect to λ_m of Eq. (11) equal to zero

$$\sum_{k=1}^{K} q\left(m\middle|x_k,\hat{\Theta}\right)\frac{\partial}{\partial\lambda_m}\log\left(p_m\left(x_k\middle|\lambda_m\right)\right) = 0. \qquad (14)$$

Putting Eq. (13) into Eq. (14) and applying logarithm-rules we get

$$\lambda_m = r_m \cdot \sum_{k=1}^{K} q\left(m\middle|x_k,\hat{\Theta}\right) \Big/ \sum_{k=1}^{K} q\left(m\middle|x_k,\hat{\Theta}\right)\cdot x_k. \qquad (15)$$

Note that Eqs. (12) and (15) together with Eq. (7) are simple closed-form expressions for the parameters of a HErD according to a given number of Erlang branches M and a given number of phases r_m per branch.

A pseudo-code representation of the EM algorithm tailored to the parameter estimation of HErD is presented in Fig. 3. Note that each iteration (see steps (2) to (7) in Fig. 3) is guaranteed to increase the log-likelihood value and the algorithm is guaranteed to converge to a local maximum of the likelihood function [15]. To check whether convergence is reached, we compute in each iteration either

(i) the maximal difference of the values of the parameter vectors of successive iterations,

(ii) the relative difference of the log-likelihood values of successive iterations,

and stop the algorithm when the computed difference is below a predefined ε, e.g. $\varepsilon = 10^{-6}$. The computational complexity of the E-step is O(M·K) when computing the nominator and denominator of the unobserved data pmf separately. The complexity of the M-step is also O(M·K). Thus, the overall computational complexity for one iteration of the EM algorithm is O(M·K). Note that the log-likelihood (see Eq. (5)) can be computed without additional effort during the E-step of the fitting algorithm.

A straightforward computation of the Erlang densities (13) can exhibit numerical difficulties, since for a high number of Erlang phases (e.g. r > 50) large factorials and large power values must be computed. To avoid these difficulties, we suggest an evaluation of Eq. (13) in logarithmic form, i.e.,

$$p_m\left(x_k\middle|\lambda_m\right) = \lambda_m e^{(r_m-1)\ln(\lambda_m x_k)-\ln(r_m-1)!-\lambda_m x_k}, \qquad (16)$$

with pre-computed logarithms of the factorial values,

$$\ln r! = \sum_{i=1}^{r}\ln i. \qquad (17)$$

On a standard PC with 3 GHz Pentium CPU running the operating system Linux, the EM algorithm requires about 2.4 seconds for 100 iterations when fitting a HErD with M = 5 Erlang branches to a trace with $K = 10^4$ samples. The overall number of iterations required to achieve convergence depends on several factors, i.e., the initial setting of α_m and λ_m, the number of Erlang branches M, and the trace data. However, for small values of M (i.e., $M \le 10$) the algorithm converges faster than for larger values of M, since fewer parameters have to be optimized. With $M \le 10$ the number of iterations is almost always less then 100 to reach convergence with $\varepsilon = 10^{-6}$.

(1) Choose initial estimates $\hat{\Theta} = (\hat{\alpha}_1,\ldots,\hat{\alpha}_M,\hat{\lambda}_1,\ldots,\hat{\lambda}_M)$

(2) **REPEAT**

(3) Compute $p_m(x_k|\hat{\lambda}_m)$ for m=1,…,M and k=1,…,K according to Eq. (13)

(4) **E-step:** Compute the pmf of the unobserved data for m=1,…,M and k=1,…,K

$$q\left(m\middle|x_k,\hat{\Theta}\right) = \hat{\alpha}_m \cdot p_m\left(x_k\middle|\hat{\lambda}_m\right) \Big/ \sum_{i=1}^{M}\hat{\alpha}_i \cdot p_i\left(x_k\middle|\hat{\lambda}_i\right)$$

(5) **M-step:** Compute α_m and λ_m that maximize Eq. (11) for m=1,…,M

$$\alpha_m = \frac{1}{K}\sum_{k=1}^{K} q\left(m\middle|x_k,\hat{\Theta}\right) \text{ and}$$

$$\lambda_m = r_m \cdot \sum_{k=1}^{K} q\left(m\middle|x_k,\hat{\Theta}\right) \Big/ \sum_{k=1}^{K} q\left(m\middle|x_k,\hat{\Theta}\right)\cdot x_k$$

(6) set $\hat{\Theta} := \Theta$

(7) **UNTIL** convergence reached according to criterion (i) or (ii)

(8) **RETURN** optimal parameter vector $\Theta = (\alpha_1,\ldots,\alpha_M,\lambda_1,\ldots,\lambda_M)$

Fig. 3. Pseudo-code of the EM algorithm tailored to hyper-Erlang distributions

4. Finding an Optimal Setting of the Discrete Parameters of a HErD

With the EM algorithm presented in Section 3.2 we can optimize the continuous parameter vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\lambda}$ of a HErD for a predefined setting of the number of Erlang branches M and number of phases of each Erlang branch r_m, m=1,…,M. However, in order to find the "best" HErD with N states we have to consider all HErD out of the set $\mathcal{H}_N$ as candidates. Due to the efficiency of the algorithm it is feasible to enumerate all possible settings of M and $r_1,\ldots,r_M$ and to fit for each such setting a HErD, if N is small (i.e, $N \leq 10$) and K is not too large (i.e., $K \leq 10^6$). Comparing the fitted HErD according to their log-likelihood values and choosing the one with the maximal log-likelihood value gives the best HErD in this case. Recall, that the log-likelihood values can be computed without additional computational effort according to Eq. (5).

Formally, we denote the *discrete parameter setting* of a HErD f(x; M, $\mathbf{r}$, $\boldsymbol{\alpha}$, $\boldsymbol{\lambda}$) by the tuple (M, $\mathbf{r}$). The following lemma provides a recursive formula to compute the overall number of settings of a HErD with N states, denoted by S_N.

Lemma 2: The overall number of different N-state settings, S_N, is given by $\varphi_N(N,0)$, where

$$\varphi_m(n,j) = \sum_{i=j}^{\lfloor n/m \rfloor} \varphi_{m-1}(n-i,i) \text{ and}$$

$$\varphi_1(n,j) = 1, \text{ for all } n \geq j \qquad (18)$$

A proof of Lemma 2 is provided in [10]. In fact, for N = 5, 6, 7, 8, 9, 10 only S_N = 7, 11, 15, 22, 30, 42 settings exist. Unfortunately, for larger N the number of settings grows exponentially, e.g., for N = 20 we have 627 different settings. Thus, for large values of N, i.e., N > 10, it is not feasible to apply the EM algorithm for every possible setting. The same holds when fitting even one setting takes some time, which may be the case for very large traces (e.g. $K > 10^7$ sample). In these cases we recommend using one of the following strategies:

(i) **Progressive pre-selection:** In a first round enumerate all possible settings and apply the EM algorithm until convergence with $\varepsilon = 10^{-3}$ is reached. This requires only a few iterations for each setting. Select the settings with the best log-likelihood values and put them into a priority queue. We commonly consider at least 5 and at most 50 settings in this round. Then start a second round with continuing iteration of the selected HErD until convergence with $\varepsilon = 10^{-4}$ is reached. Finally, start a third round with the 50% best of the priority queue until $\varepsilon = 10^{-6}$. Experimental results when applying this strategy are presented in Section 5.

(ii) **Special structures:** If the empirical distribution of the trace has a small squared coefficient of variation (i.e., $c^2 < 1$) we recommend to fit the HErD only with one, two, or three Erlang branches, i.e., M = 1, M = 2, or M = 3. Note that the number of N-state settings with $M \leq M_{max}$ Erlang branches is $\varphi_{M_{max}}(N,0)$, e.g., for N = 50 and M_{max} = 2 only 26 settings must be fitted. For monotonically decreasing empirical distributions with large squared coefficient of variation (i.e., $c^2 > 1$) we recommend to fit the HErD only with N, N-1, or N-2 Erlang branches. Note that M = N corresponds to a hyperexponential distribution, which was shown to fit heavy-tailed distributions with large squared coefficient of variation quite well in [9].

(iii) **Body/tail fitting:** Fit the body of a distribution with a (say) 10-state HErD with M = 1, 2, and 3 Erlang branches, which requires only 1+5+8 = 14 runs of the EM algorithm. Fit the tail of distribution with an (say) 8-state hyper-exponential distribution, i.e., M = 8 and $r_1 = \ldots = r_M = 1$. Apply this combined body/tail fitting on an 18-state HErD. Thus, an overall number of 14 settings must be evaluated. A good application example for this approach is the Pareto-II distribution discussed in Section 5.

The first approach works automatically, but requires additional effort, which is not required if the other two variants are used. In practice especially variant (ii) works well, but requires some pre-analysis of the data and an experienced user to decide about a good range for the discrete parameters. The separate fitting of body and tail, as suggested in (iii), is often used for heavy tailed distributions (e.g., [18]), but requires an appropriate definition of body and tail, and the number of states used for their approximation. Additionally, the low complexity of the presented HErD fitting method allows us to optimize the body and the tail fitting parameters together, which is preformed separately in [12]. The results from [7] indicate that a common fitting algorithm might yield excellent results for fitting heavy tailed distributions.

5. Experimental Results

5.1 Synthetically Generated Traces

In the experiments a hyper-Erlang distribution (HErD) and an acyclic phase-type distribution (APHD) are fitted for given traces with 10^4 samples drawn from

known distributions. In particular we consider two Weibull distributions with scale parameter $\eta = 1.0$ and shape parameter $\beta = 0.5$ and $\beta = 5.0$, respectively, and a uniform distribution with left and right boundary equal to 0.5 and 1.5. In addition we consider a Pareto-like distribution with heavy tail index $\alpha = 1.5$ and $b = 2.0$. This distribution was previously used in [11] as an example of a heavy tailed distribution, which is not monotonically decreasing. According to [11] it is denoted a Pareto-II distribution. Furthermore, we consider the shifted exponential distribution as well as the matrix exponential distribution, which are part of a set of benchmark distributions for PH fitting algorithms defined in [3]. The non-standard density functions are summarized in Tab. 1.

Tab. 1. Densities of considered distributions

Pareto-II(α,b):	$f_{paretoII}(x;\alpha,b) = \frac{b^{\alpha} e^{-b/x}}{\Gamma(\alpha)} x^{-\alpha-1}$
Shifted exp. (SE):	$f_{SE}(x) = \begin{cases} \frac{1}{2} e^{-x} & ,0 \le x < 1 \\ \frac{1}{2} e^{-x} + \frac{1}{2} e^{-(x-1)} & ,x \ge 1 \end{cases}$
Matrix exp. (ME):	$f_{ME}(x) = \left(1 + \frac{1}{(2\pi)^2}\right) \cdot (1 - \cos(2\pi x)) \cdot e^{-x}$

With respect to the set of distributions we compared the fitting quality of the HErD found by G-FIT with the quality of the APHD found by the tool PH-fit [12]. The PH-fit tool approximates the optimal parameter set of an APHD by minimizing a predefined distance measure with a non-linear optimization algorithm. This algorithm uses an iterative linearization method based on numerical computation of partial derivatives and the simplex method to determine the direction in which the distance measure decreases most. In the presented comparison the fitting parameters of the PH-fit tool were the following: only body fitting is applied (i.e., no separate tail fitting) and the distance of the original and the approximate distributions is calculated up to the largest sample value. Furthermore, we run PH-fit with 3 rounds (i.e., starting from 3 different initial guesses) and at most 200 modifications in each round.

Fig. 4 shows the empirical density functions for the six traces as well as the density functions for the fitted HErD and APHD with N = 5 states and N = 10 states, respectively. For some of the distributions also results when fitting a HErD with 50 states are plotted. Densities of traces are approximated by histograms with intervals of width 0.05. The results for G-FIT are obtained by fitting a HErD with the algorithm of Fig. 3 for all possible discrete parameter settings. Recall, that for N = 5 only 7 settings and for N = 10 only 42 settings are considered. The EM algorithm stops when convergence is reached according to the log-likelihood criterion with $\varepsilon = 10^{-6}$ (see criterion (ii) in Section 3.2). From the curves of Fig. 4 we conclude that the fitting quality of HErD is almost always as good as the quality for APHD. Moreover, in some cases the results for HErD are better than for APHD, e.g. when fitting the uniform distribution with 5 states. The reason why PH-fit did not find the best solution in these cases is that the optimization process got stuck in a local optimum.

Tab. 2 presents several quality indices for the considered distributions. In particular, the first three moments and the squared coefficient of variation for each of the six traces as well as the fitted HErD and APHD are presented. Relative errors of the fitted distributions are presented in brackets behind the absolute values. Furthermore, Tab. 2 contains for each trace the log-likelihood value and the CPU time required by G-FIT and PH-FIT. In the last row of each distribution the optimal Erlang phase lengths r_m found by G-FIT are shown. Recall, that for N = 5 and N = 10 the G-FIT results are found from the best fit when fitting a HErD for all possible discrete parameter settings. Applying the progressive pre-selection (see strategy (i) in Section 4) yields almost always the same results but requires less CPU time (see numbers in brackets in the rows with CPU time in Tab. 2). In fact, only for the shifted exponential trace (log-likelihood –13280.73) the results are not as good as in the general case (indicated with an asterisk behind the brackets in Tab. 2). The reason for this is that with progressive pre-selection some settings may be canceled out of the priority queue in the first or second round which would get better when running the EM algorithm until convergence with $\varepsilon = 10^{-6}$.

Results presented in Tab. 2 when applying G-FIT with 20 states are computed with progressive pre-selection. Note, that not all of the $\varphi_{20}(20,0) = 627$ settings are evaluated, but only a part of them which seems to be reasonable (see strategy (ii) in Section 4). In fact, for the Weibull(1.0,0.5) trace we considered all settings with $M \ge 12$ Erlang branches (67 settings), for the Weibull(1.0,5.0) trace and the matrix exponential trace we considered all settings with $M \le 5$ Erlang branches (192 settings), and for the Pareto-II trace and the shifted exponential trace we fitted all settings with $M \le 6$ Erlang branches (282 settings). Comparing the fitted HErD and APHD, it can be observed that for all distributions the fitting quality of the 20-state HErD is much better than that of the 10-state APHD. Moreover, the fitting process for the 20-state HErD is less time consuming as for the 10-state APHD, although the number of states is doubled.

Tab. 2. Quality indices of fitted HErD and APHD for synthetically generated traces

Quality indices		Trace	5 states		10 states		20 states	Special cases
			G-FIT	PH-FIT	G-FIT	PH-FIT	G-FIT	G-FIT
1. Moment	Weibull(1.0,0.5)	1.99	1.99 (0.0%)	1.98 (0.5%)	1.99 (0.0%)	1.91 (4.3%)	1.99 (0.0%)	1.99 (0.0%)
2. Moment		25.61	22.47 (12.3%)	20.98 (18.1%)	23.78 (7.1%)	17.77 (30.6%)	24.26 (5.3%)	25.76 (0.6%)
3. Moment		995.13	512.55 (48.5%)	422.44 (57.5%)	638.04 (35.9%)	300.78 (69.8%)	701.43 (29.5%)	1156.57 (16.2%)
Squared CoV		5.45	4.66 (14.5%)	4.34 (20.5%)	4.99 (8.4%)	3.89 (28.6%)	5.11 (6.2%)	5.49 (0.7%)
Log-likelihood			-11068.87	-11364.43	-10998.60	-11258.91	-10986.43	-10992.01
CPU time [sec]			13 (12)	249	170 (72)	461	268	55
Phase lengths			1,1,1,1,1		1,1,1,1,1,1,1,1,1,1		1,...,1,2,2	1,1,1,1,1,1,1,1
1. Moment	Uniform(0.5,1.5)	1.00	1.00 (0.0%)	1.00 (0.1%)	1.00 (0.0%)	1.00 (0.0%)	1.00 (0.0%)	1.00 (0.0%)
2. Moment		1.08	1.19 (10.6%)	1.24 (15.1%)	1.09 (1.4%)	1.09 (1.4%)	1.08 (0.6%)	1.08 (0.7%)
3. Moment		1.24	1.66 (33.9%)	1.85 (49.2%)	1.30 (5.2%)	1.31 (5.3%)	1.27 (2.7%)	1.27 (2.2%)
Squared CoV		0.08	0.20 (135.7%)	0.25 (194.6%)	0.10 (17.9%)	0.10 (18.3%)	0.09 (7.1%)	0.09 (8.6%)
Log-likelihood			-3105.56	-3804.21	-1852.51	-1855.00	-1827.56	-1166.51
CPU time [sec]			2 (1)	182	28 (13)	386	137	13
Phase lengths			5		10		2,2,2,3,11	21,29
1. Moment	Shifted Exponential	1.51	1.51 (0.0%)	1.51 (0.1%)	1.51 (0.0%)	1.51 (0.0%)	1.51 (0.0%)	1.51 (0.0%)
2. Moment		3.58	3.61 (1.0%)	3.59 (0.3%)	3.61 (0.8%)	3.61 (1.0%)	3.59 (0.3%)	3.53 (1.5%)
3. Moment		11.55	11.96 (3.5%)	11.69 (1.2%)	11.79 (2.1%)	12.47 (7.9%)	11.59 (0.3%)	11.01 (4.7%)
Squared CoV		0.56	0.58 (2.7%)	0.57 (1.1%)	0.58 (2.1%)	0.58 (2.7%)	0.57 (0.8%)	0.54 (4.0%)
Log-likelihood			-13376.14	-13327.48	-13265.52	-13253.28	-13184.11	-13148.89
CPU time [sec]			5 (3)	140	90 (20)*	205	116	309
Phase lengths			1,1,3		1,3,6		1,1,2,4,12	1,13,17,19

		Trace	5 states		10 states		20 states	Special cases
			G-FIT	PH-FIT	G-FIT	PH-FIT	G-FIT	G-FIT
	Weibull(1.0,5.0)	0.92	0.92 (0.0%)	0.92 (0.1%)	0.92 (0.0%)	0.92 (0.0%)	0.92 (0.0%)	0.92 (0.0%)
		0.89	1.02 (14.1%)	1.02 (13.9%)	0.93 (4.6%)	0.93 (4.5%)	0.90 (1.1%)	0.89 (0.1%)
		0.90	1.31 (45.6%)	1.31 (45.3%)	1.03 (14.4%)	1.03 (14.4%)	0.93 (3.6%)	0.91 (0.4%)
		0.05	0.20 (283.9%)	0.20 (284.1%)	0.10 (91.9%)	0.10 (92.3%)	0.06 (21.2%)	0.05 (2.1%)
			-1673.19	-1674.49	394.68	391.10	1092.56	1402.30
			2 (1)	217	27 (14)	434	139	34
			5		10		1,2,17	16,34
	Pareto-II(1.5,2.0)	4.34	4.34 (0.0%)	3.89 (10.3%)	4.34 (0.0%)	3.78 (12.9%)	4.34 (0.0%)	4.34 (0.0%)
		1057.62	323.6 (69.4%)	117.6 (88.9%)	340.7 (67.8%)	105.9 (90.0%)	911.7 (13.8%)	978.0 (7.5%)
		1768568	100147 (94.3%)	12348 (99.3%)	114715 (93.5%)	12579 (99.3%)	1389436 (21.4%)	1674493 (5.3%)
		55.27	16.21 (70.7%)	6.78 (87.7%)	17.12 (69.0%)	6.42 (88.4%)	47.50 (14.1%)	51.03 (7.7%)
			-20340.91	-20236.15	-20084.11	-20108.91	-19980.79	-20093.74
			2 (2)	246	62 (23)	423	156	29
			1,1,3		1,2,3,4		1,1,2,3,6,7	1,1,1,1,2,4
	Matrix Exponential	1.06	1.06 (0.0%)	1.06 (0.0%)	1.06 (0.0%)	1.06 (0.2%)	1.06 (0.0%)	1.06 (0.0%)
		2.12	2.15 (1.7%)	2.10 (0.7%)	2.16 (1.9%)	2.10 (1.1%)	2.13 (0.4%)	2.13 (0.5%)
		6.57	6.66 (1.2%)	6.20 (5.6%)	7.24 (10.2%)	6.11 (7.0%)	6.59 (0.2%)	6.53 (0.7%)
		0.89	0.92 (3.7%)	0.88 (1.4%)	0.92 (4.1%)	0.86 (3.3%)	0.90 (0.9%)	0.90 (1.0%)
			-9278.27	-9230.53	-8895.56	-8778.50	-8748.19	-8581.07
			4 (2)	142	60 (17)	285	107	174
			2,3		1,4,5		1,5,6,8	5,12,15,18

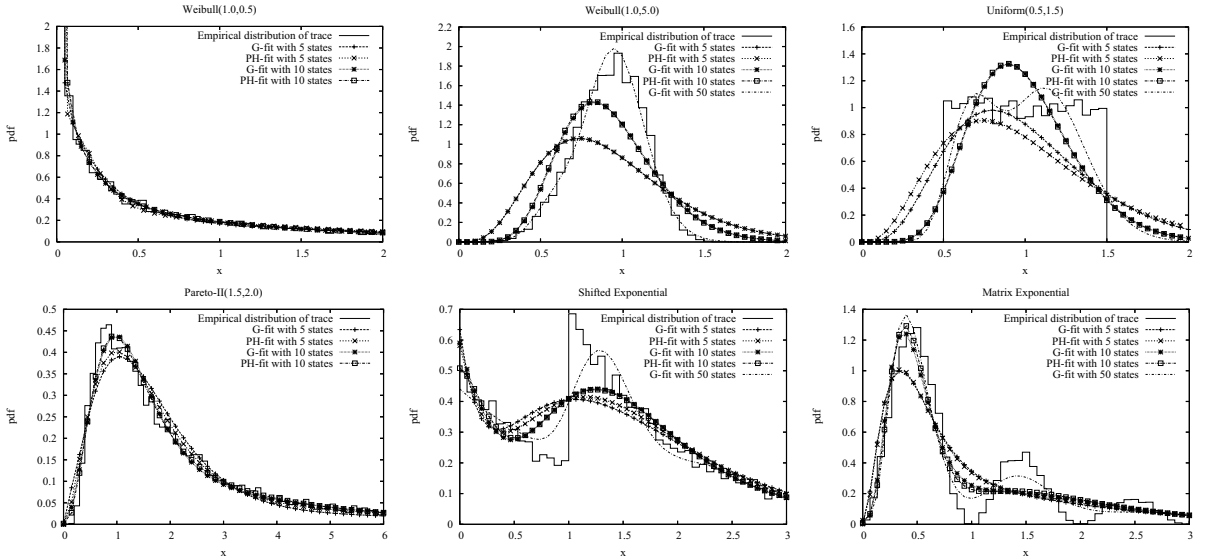

Fig. 4. Densities of fitted HErD and APHD for synthetically generated traces

The last column in Tab. 2 shows results for G-FIT for some special cases. In particular, for the Weibull(1.0,0.5) trace results for an 8-state hyperexponential distribution running the EM algorithm until convergence with $\varepsilon = 10^{-8}$ are presented. It can be observed that due to the longer iteration time the moments are matched much better than in the case when stopping the iteration with $\varepsilon = 10^{-6}$. For the Weibull(1.0,5.0) trace and the uniform trace we fitted all 50-state settings with at most two Erlang branches (26 settings) until convergence is reached with $\varepsilon = 10^{-16}$. The time requirements for the fitting process are still very small as can be observed from Tab. 2. For the shifted exponential trace and the matrix exponential trace we applied progressive pre-selection with 50 states and $M \leq 4$ Erlang branches (1154 settings). Finally, for the Pareto-II trace we applied the body/tail fitting approach (see strategy (iii) in Section 4). We used 6 states for the body and 4 states for the tail. For the body we fitted all settings with two Erlang branches (3 settings) until convergence with $\varepsilon = 10^{-10}$. As expected, the first three moments are fitted very well with this approach and the tail-behavior of the distribution is captured much better, although the log-likelihood value is worse compared to the best 10-state HErD.

Fig. 5 depicts the tail-behavior for the Pareto-II distribution when fitting a trace with $K = 10^6$ samples. Recall that CPU time requirements are linear in the size of the trace, i.e., for a trace with 10^6 samples the time requirements presented in Tab. 2 must be roughly multiplied by 100. Note that the quality of tail fitting in Fig. 5 is also a function of the number of used samples. The same HErD structures would result in a longer tail fit with more samples.

In a final experiment, we compared the results obtained by G-FIT with results from the PH fitting tool EMpht [1]. Similar to G-FIT the tool EMpht applies

the EM algorithm for distribution fitting, but with no specialization to a sub-class of PH distributions. Throughout all experiments G-FIT outperforms EMpht in terms of CPU time requirements. Furthermore, EMpht converges much slower to optimal parameter values than G-FIT. For example, for the Weibull(1.0,0.5) trace EMpht required for 1000 iterations on a 5-state PH distribution 260 seconds CPU time and reached only a log-likelihood value of –11481.29 which is worse than that for G-FIT and also PH-FIT (see Tab. 2). For the uniform trace EMpht required for 1000 iterations on a 10-state PH distribution 230 seconds CPU time and reached a log-likelihood value of –2034.95, which is also worse than that for G-FIT and PH-FIT. Fitting the Pareto-II trace with a 10-state distribution seems to be not practicable since already 100 iterations take more than 270 seconds of CPU time with log-likelihood values still far away from the optimum.

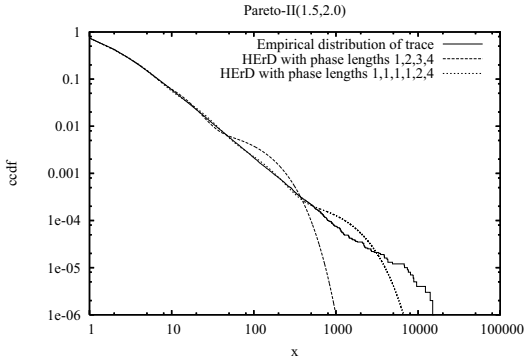

Fig. 5. Complementary cdf for two different fitted HErD for the Pareto-II trace

The results presented in this section underline the flexibility of the class of HErD for fitting general distributions as theoretically shown in Section 2. Furthermore, we conclude from the experiments that HErD can be fitted much more efficiently and in most cases more accurately than APHD with the proposed EM algorithm. We believe that this is essentially due to the more restricted structure of the HErD class which practically does not reduces its flexibility on fitting. We think that other fitting algorithms over the HErD class would result in similar fitting quality but are less efficient in terms of CPU time requirements.

5.2 Real Traffic Traces

To study an example with a real data traffic trace we used the call center data trace provided by Avishai Mandelbaum [17]. The data archives all calls handled by the call center of one of Israel's banks over a period of 12 month from January 1999 till December 1999. For every month about 20,000 to 30,000 calls are recorded. For every call the traces contain several attributes, from which we used the service times as given in the traces for our study. Furthermore, service times are scaled to have mean 1.0.

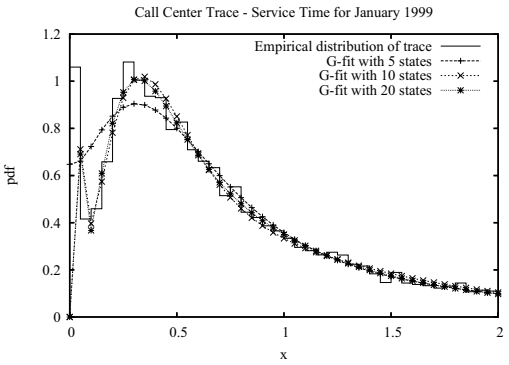

Fig. 6. Densities for call center trace

Fig. 6 shows the empirical density functions for the trace of January as well as the density functions for the fitted HErD with 5, 10, and 20 states, respectively. Service times exhibited a quick-hang phenomena [17], i.e., there is a high percentage of calls with very short service times. From Fig. 6 we conclude that the trace is fitted very well by a HErD with 10 or 20 states whereas 5 states seem to be not sufficient to adequately represent the traces. Furthermore, Fig. 6 shows that the quick-hang phenomena can also be represented by the HErD with 10 states or 20 states. The relative difference between the moments of the trace and the moments of the 20-state HErD is at most 1%, which underlines the high accuracy of the fitted distribution.

Tab. 3. Quality indices for LBL-TCP-3 trace

	Trace	5 states	10 states
1. Moment	1.00	1.00 (0.0%)	1.00 (0.0%)
2. Moment	2.94	2.87 (2.3%)	2.92 (0.9%)
3. Moment	16.84	14.94 (11.3%)	15.99 (5.0%)
Squared CoV	1.94	1.87 (3.5%)	1.92 (1.3%)
Log-likelihood		-1672401.67	-1665271.83
CPU time [sec]		286	1670
Phase lengths		1,2,2	2,2,2,2,2

To provide a second example we considered the LBL-TCP-3 trace from the internet traffic archive [13]. The trace contains about $1.8 \cdot 10^6$ TCP timestamps from which we extracted the interarrival times of TCP packets and scaled the data set to have mean 1.0. The empirical distribution of the trace and the fitted HErD with 5 and 10 states are presented in Fig. 7. Furthermore, Tab. 3 shows statistical properties concerning the trace and the fitted distributions. The CPU time requirements are measured when applying progressive pre-selection for finding the best setting of the discrete parameters of the HErD. From the results we conclude that even very large traces (i.e., 10^6 – 10^8

samples) can be fitted efficiently and accurately with the proposed method.

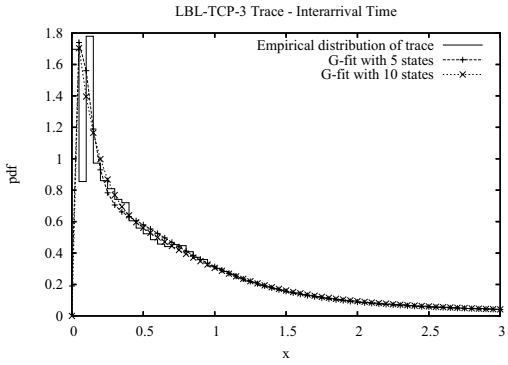

Fig. 7. Densities for LBL-TCP-3 trace

6. Conclusions

We presented a novel approach that fits a restricted class of phase-type distributions to trace data. For the parameter fitting we developed an EM algorithm, which is tailored to the special structure of a hyper-Erlang distribution. One of the crucial ideas behind the fitting method presented in this paper is the use of the smallest class of phase-type distributions, which is still sufficiently general to approximate any non-negative distribution (see Theorem 1 and Fig. 2). The empirical experiences confirm the expectation that searching for best fitting in a smaller class of distributions is numerically more effective and stable.

The effectiveness of the proposed fitting method is demonstrated by a comparison with two other methods using six benchmark traces and two real traffic traces. We conclude from this comparison that hyper-Erlang distributions are the most versatile sub-class of phase-type distributions, since hyper-Erlang distributions provide practically the full flexibility of the PH class and can be efficiently tuned to match general distributions.

References

[1] S. Asmussen, O. Nerman, and M. Olsson, Fitting Phase-type Distributions via the EM Algorithm, *Scandinavian Journal of Statistics* **23**, 419-441, 1996. `http://www.maths.lth.se/matstat/staff/asmus/pspapers.html`.

[2] A. Bobbio, A. Horváth, M. Scarpa, and M. Telek, Acyclic discrete phase type distributions: Properties and a parameter estimation algorithm, *Performance Evaluation* **54**, 1-32, 2003.

[3] A. Bobbio and M. Telek, A Benchmark for Estimation Algorithms: Results for Acyclic-PH, *Stochastic Models* **10**, 661-677, 1994.

[4] P. Buchholz, An EM-Algorithm for MAP Fitting from Real Traffic Data, *Proc. 13th Int. Conf. on Modelling Tools and Techniques for Computer and Communication System Performance Evaluation, LNCS* **2794**, 218-236, 2003.

[5] A. Cumani, On the Canonical Representation of Homogeneous Markov Processes Modeling Failure – Time Distributions, *Journal on Microelectronics and Reliability* **22**, 583-602, 1982.

[6] A.P. Dempster, N.M. Laird, and D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, *Journal of the Royal Statistical Society* **39**, 1-38, 1977.

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

[8] Y. Fang, Hyper-Erlang Distribution Model and its Application in Wireless Mobile Networks, *Wireless Networks* **7**, 211-219, 2001.

[9] A. Feldmann and W. Whitt, Fitting mixtures of exponentials to long-tail distributes to analyze network performance models, *Performance Evaluation* **31**, 245-258, 1998.

[10] G-FIT, `http://ls4-www.cs.uni-dortmund.de/home/thummler/pubs.html`.

[11] A. Horváth and M. Telek, Markovian Modeling of Real Data Traffic: Heuristic Phase Type and MAP Fitting of Heavy Tailed and Fractal Like Samples, *Performance 2002*, *LNCS* **2459**, 405-434, 2002.

[12] A. Horváth and M. Telek, Phfit: A general phase-type fitting tool, *Proc. 12th Int. Conf. on Modelling Tools and Techniques for Computer and Communication System Performance Evaluation, London, UK, LNCS* **2324**, 82-91, 2002. `http://webspn.hit.bme.hu/~telek/tools.htm`.

[13] Internet Traffic Archive, `http://ita.ee.lbl.gov/index.html`.

[14] M.A. Johnson, Selecting Parameters of Phase Distributions: Combining Nonlinear Programming, Heuristics, and Erlang Distributions, *ORSA Journal on Computing* **5**, 69-83, 1993.

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

[16] A. Lang and J. L. Arthur, Parameter Approximation for Phase-Type Distributions. Matrix Analytical Methods in Stochastic Models, *Lect. Notes in Pure and Applied Math.* **183**, 151-206, Marcel Dekker, 1997.

[17] A. Mandelbaum, A. Sakov, and S. Zeltyn, Empirical analysis of a call center, *Tech. Rep., Technion, Israel Institute of Technology*, 2000. `http://iew3.technion.ac.il/serveng/callcenterdata`.

[18] A. Riksa, V. Diev, and E. Smirni, An EM-based technique for approximating long-tailed data sets with PH distributions, *Performance Evaluation* **55**, 147-164, 2004.

[19] L. Schmickler, MEDA: Mixed Erlang Distributions as Phase-Type Representations of Empirical Distribution Functions, *Stochastic Models* **8**, 131-156, 1992.

[20] K.S. Trivedi, Probability and Statistics with Reliability, Queuing and Computer Science Applications, Second Edition, John Wiley & Sons, 2002.