Fitting Mixtures of Exponentials to Long-Tail Distributions to Analyze Network Performance Models

Anja Feldmann Ward Whitt
AT&T Laboratories – Research
Murray Hill, NJ 07974-0636, USA
{anja,wow}@research.att.com

Abstract

Traffic measurements from communication networks have shown that many quantities characterizing network performance have long-tail probability distributions, i.e., with tails that decays more slowly than exponentially. Long-tail distributions can have a dramatic effect upon performance, but it is often difficult to describe this effect in detail, because performance models with component long-tail distributions tend to be difficult to analyze. We address this problem by developing an algorithm for approximating a long-tail distribution by a finite mixture of exponentials. The fitting algorithm is recursive over time scales. At each stage, an exponential component is fit in the largest remaining time scale and then the fitted exponential component is subtracted from the distribution. Even though a mixture of exponentials has an exponential tail, it can match a long-tail distribution in the regions of primary interest when there are enough exponential components.

1 Introduction

A major challenge for engineering the emerging high-speed integrated-services communication networks is to develop models that can realistically capture the performance effects of the complex traffic that will be offered to and carried by these networks. Evidence of traffic complexity appears in many forms, such as in the long-range dependence and self-similarity found in the statistical analysis of traffic measurements [11]. There is also strong evidence of important phenomena at several different time scales [13].

In this paper we focus on one phenomenon that seems to underlie much of the observed traffic complexity: long-tail probability distributions. Let F be a *cumulative distribution function* (cdf) and let the associated *complementary cdf* (ccdf) be $F^c(t) = 1 - F(t)$. We say that a cdf F (or its associated ccdf F^c) has a *long tail* (also known as fat tail or heavy tail) if the ccdf F^c decays more slowly than exponentially, i.e., if

$$e^{\gamma t} F^c(t) \to \infty \quad \text{as} \quad t \to \infty \quad \text{for all } \gamma > 0 \ . \tag{1.1}$$

In contrast, we say that a cdf F has a *short tail* if its ccdf F^c decays exponentially, i.e., if there exists some $\gamma > 0$ such that

$$e^{\gamma t} F^c(t) \to 0 \quad \text{as } t \to \infty \ . \tag{1.2}$$

Neither (1.1) nor (1.2) describes the actual decay rates of the ccdf's well; they are intended for general classification. A typical long-tail cdf might have a *power tail*, i.e.,

$$F^c(t) \sim \alpha t^{-\beta} \quad \text{as} \quad t \to \infty \ , \tag{1.3}$$

where α and β are positive constants and $f(t) \sim g(t)$ as $t \to \infty$ means that $f(t)/g(t) \to 1$ as $t \to \infty$, whereas a typical short-tail cdf might have *bounded support* ($F^c(t) = 0$ for some t) or an *exponential tail*, i.e.,

$$F^c(t) \sim \alpha e^{-\eta t} \quad \text{as} \quad t \to \infty \tag{1.4}$$

for positive constants α and η.

Two familiar long-tail distributions are the Pareto distribution and the Weibull distribution. One form of the *Pareto distribution*, which we refer to as Pareto(a,b), has ccdf

$$F^c(t) = (1 + bt)^{-a} \tag{1.5}$$

for positive parameters a and b (p. 233 [10]). One form of the *Weibull distribution*, which we refer to as Weibull(c, a), has ccdf

$$F^c(t) = e^{-(t/a)^c} \tag{1.6}$$

for positive parameters a and c (Ch. 20 [10]). From (1.5) it is easy to see that the Pareto ccdf in (1.5) has a power tail and so always has a long tail. The Weibull ccdf in (1.6) has a long tail but not a power tail if $c < 1$.

In recent years, many traffic measurement studies have found long-tail distributions. For example, the analysis of a large dataset of local area Internet IP traffic collected at Bellcore showed that traffic is highly variable over several time scales. Measurements of source on and off times (high and low activity times) of individual network sources within the Bellcore dataset have indicated long-tail distributions [11], and it has been proven that such long-tailed on and off times for individual sources can explain the self-similarity in the aggregate traffic [19].

Long-tail distributions yield statistically better models for the tail behavior of durations, number of bytes, and burst bytes of ftp connections on the Internet [15]. Intervals

0-8186-7780-5/97 $10.00 © 1997 IEEE

between connection requests in Internet traffic have long-tail distributions [7]. Recent analysis of the durations of world wide web transfers have led to scrutinizing the file length distribution on file servers [6]. Both distributions have been found to be long-tailed.

The accumulated evidence is clear: many important probability distributions associated with network traffic have long tails. Moreover, it is known that long-tail distributions can have a dramatic impact upon network performance. For example the steady-state waiting-time distribution in an infinite-capacity single-server queue inherits the long-tail property of a service-time distribution. However, the impact of a long-tail distribution depends on the context and requires careful analysis. For example, in the single-server queue, large delays are caused by large service times and short interarrival times.

Not only are long-tail distributions prevalent and important, but they are difficult to analyze. For example, even the relatively simple $M/G/1$ queue is difficult to analyze when the service-time distribution is Pareto. Abate et al. [1] calculate performance measures for the $GI/G/1$ queue when the general interarrival-time and service-time distributions are long-tailed using numerical transform inversion, but it is necessary to have the Laplace transforms of these distributions, and there evidently is no convenient expression for the Laplace transforms of the Pareto and Weibull distributions.

Our main contribution in this paper is to point out that it is possible to approximate long-tail probability distributions by convenient short-tail probability distributions, so that available performance models can be effectively analyzed and so that the effect of the long-tail distribution upon performance can be determined. (We do *not* claim that the long-tail distribution has no effect.) Moreover, we develop a remarkably simple algorithm for constructing suitable approximating distributions for a large class of long-tail distributions. The class of long-tail distributions that can be approximated by the method developed here includes the Pareto and Weibull distributions in (1.5) and (1.6) as special cases.

Although at first it may be surprising that long-tail distributions can be approximated by short-tail distributions, there is a simple explanation in the notion of time scale. In almost all network performance settings, the distribution of interest only matters through its values in some finite interval $[t_1, t_2]$. For t_1 sufficiently small and t_2 sufficiently large, the precise form of the distribution outside the interval $[t_1, t_2]$ should not matter. (Because of the nature of time scales, it is usually appropriate to measure time logarithmically. Thus, we might have $t_1 = 10^{-a}$ and $t_2 = 10^b$ for appropriate constants a and b.) The main point is that, in principle, it should be possible to approximate any long-tail distribution by a short-tail distribution. A simple way to do this is to truncate the distribution at the points t_1 and t_2 and assign the negligible probabilities of the intervals $[0, t_1)$ and (t_2, ∞) to the points t_1 and t_2, respectively. Although this produces a short-tail distribution that captures the essential behavior of the original long-tail distribution, it may not be a convenient approximation.

Here we consider hyperexponential distributions as approximating distributions. A *hyperexponential* (H_k) distribution is a mixture of k exponentials for some k, i.e., the ccdf has the form

$$H^c(t) = \sum_{i=1}^{k} p_i e^{-\lambda_i t} \qquad (1.7)$$

where $p_i \geq 0$ for all i and $p_1 + \ldots + p_k = 1$. Our fitting algorithm fits a hyperexponential distribution to a given long-tail distribution, aiming to be accurate over a finite interval $[t_1, t_2]$ for suitably small t_1 and suitably large t_2.

Given data that might be well described by either a Pareto distribution or a hyperexponential distribution, we would usually prefer the Pareto distribution for a simple description because it provides a more parsimonious description. The H_k distribution in (1.7) has $2k - 1$ parameters, whereas the Pareto distribution has only 2. Statistical estimation also tends to work better when there are fewer parameters. We primarily suggest replacing long-tail distributions such as the Pareto distribution by hyperexponential distributions, because performance models tend to be easier to analyze when component distributions in the model are hyperexponential. One reason is that hyperexponential distributions are special phase-type distributions, which have been found to make performance models more tractable [14]. Another reason that we might choose hyperexponential distributions is because they have simple Laplace transforms. the Laplace transform of the density h of the ccdf H^c in (1.7) and the Laplace-Stietjes Transform of the cdf H is

$$\hat{h}(s) = \int_0^\infty e^{-st} dH(t) = \sum_{i=1}^{k} \frac{p_i \lambda_i}{\lambda_i + s}. \qquad (1.8)$$

The explicit Laplace transform (1.8) makes it possible to analyze many performance models by numerical transform inversion, e.g., see [1, 5]. For these numerical transform inversion algorithms, having a relatively large number of phases (e.g., 10 or 100) presents no serious difficulty. We will illustrate this advantage by considering the $M/G/1$ queue with a long-tail service-time distribution. We have no difficulty calculating the steady-state waiting-time distribution in the $M/G/1$ queue by numerical transform inversion after making the hyperexponential approximation.

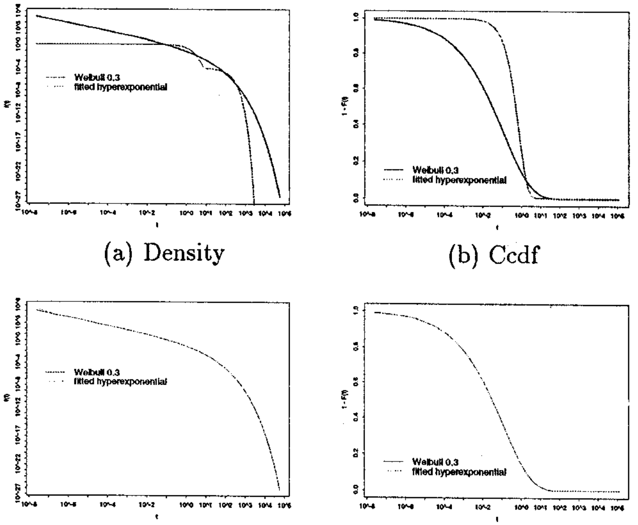
Hyperexponential distributions also make it easier to obtain Markov stochastic processes, which tend to be far easier to analyze than non-Markov stochastic processes. In particular, hyperexponential approximations can help analyze superpositions of independent on-off sources, where each source sends input at a constant rate (fluid) or as a Poisson process when it is on. If the on or off periods have long-tail distributions, then the aggregate input model tends to be intractable, but if the on and off periods of each source have hyperexponential distributions, then the aggregate input becomes a Markov-modulated fluid or Poisson process (see the longer version of this paper for details), for which there are effective algorithms. Unfortunately, however, this representation is not totally satisfactory, because the Markovian state space becomes larger when the number of exponential components in a mixture increases. Hence, if there are many sources, the state space of the approximating aggregate input model may be so large that analysis remains difficult. Nevertheless, the approximation is a step towards tractable models. If there are only a few source, then the model can now be solved, whereas it could not be solved before.

Once a hyperexponential fit is contemplated, there are many ways to proceed, such as a least squares fit using a mathematical program. A natural alternative is the expectation-maximization (EM) algorithm, which is an iterative procedure that minimizes the Kullback-Leibler "distance"; see Asmussen, Nerman and Olsson [2], Turin [17] and references therein. A difficulty with the EM algorithm is that the iteration can be slow when there are many parameters. The EM algorithm can be enhanced significantly if a good starting point can be provided. In preliminary experiments we have found that our algorithm is also useful to quickly provide a good starting point for the EM algorithm, but we do not discuss those experiments here.

We intend to compare various fitting schemes in a future paper. In this paper we present a simple recursive scheme, based on the notion of time scales. We recursively fit starting in the largest time scale that matters and successively reduce the time scale. We start by fitting a weighted exponential $p_1 e^{-\lambda_1 t}$ to the tail of the given ccdf. Since we focus on the tail, λ_1^{-1} should be suitably large. Then we subtract this weighted exponential from the original ccdf and fit a second weighted exponential $p_2 e^{-\lambda_2 t}$ to the new tail where $\lambda_2^{-1} < \lambda_1^{-1}$. Since the exponential ccdf's are short tailed, it should be possible to choose the second exponential component so that it is negligible further out in the region where the first exponential $p_1 e^{-\lambda_1 t}$ was fit. We describe the algorithm in more detail and discuss previous related work in Section 4.

To illustrate right away, we consider an example. More examples are given in an expanded version of this paper to appear in *Performance Evaluation.*

Example 1. Suppose that we consider a Weibull distribution as in (1.6) with exponent $c = 0.3$ and a chosen so that the distribution has mean 1. (That makes $a = 9.26053$.) Since c is close to 0, this Weibull distribution is strongly long-tailed. This is partly reflected by its next two moments, which are $m_2 = 29.2$ and $m_3 = 4481$. However, the first three moments do not nearly capture the full long-tail effect. To illustrate, we first consider fitting an H_2 distribution (a mixture of two exponentials, which has three parameters) to the Weibull distribution by matching the first three moments (p. 136 [18]). The resulting H_2 parameters are $p_1 = 0.00501$, $\lambda_1 = 0.019$, and $\lambda_2 = 1.355$. The approximating H_2 density and ccdf are compared to their Weibull counterparts in Figure 1 (a), (b). It is obvious that the fit is quite poor, even though the H_2 distribution has the same first three moments.



(a) Density

(b) Ccdf

(c) Density

(d) Ccdf

Figure 1: A comparison between the Weibull(0.3, 9.261) density and ccdf with hyperexponential approximations. This example shows the difference in the quality of fit between matching three moments (a), (b) and applying our algorithm (c), (d).

In contrast, the density and ccdf of an H_k fit obtained by our algorithm in Section 4 is shown in (c) and (d) of Figure 1. The fit is so good that it is hard to see two curves in (c) and (d). This H_k fit has $k = 20$ exponentials. The three moments of the approximating H_{20} distribution are $m_1 = 1.0060$, $m_2 = 30.6$, and $m_3 = 4640$.

By this example, we do not mean to imply that 20 exponentials are necessarily required to produce a satisfactory approximation of this Weibull distribution, but this number certainly seems to be sufficient for almost all network performance applications.

An attractive feature of our algorithm is that it does not depend on the moments. Therefore, it can be used even if the moments do not exist or are not known. However, it is useful to calculate the first few moments of the original and the approximating distributions to help judge the quality of the fit.

Here is how the rest of this paper is organized. In Section 2 we discuss robustness of performance models. We refer to some of the evidence indicating that if a component probability distribution in a performance model is well approximated by another, then the performance measures of interest will be suitably close. We also give a precise meaning for "close." In Section 3 we rigorously prove that it is possible to approximate many long-tail distributions by hyperexponential distributions. We identify a class of distributions containing many long-tail distributions, including Pareto and Weibull, for which arbitrarily close hyperexponential approximations can be made.

We present our recursive algorithm for constructing approximating hyperexponential distributions in Section 4. In Section 5 we explain when the algorithm should be effective.

In Section 6 we investigate how our fitting algorithm is related to fitting probability distributions to data. We show through simulation experiments that, consistent with intuition, it is usually much better to fit a long-tail distribution with only a few parameters to the data and then afterwards apply our algorithm to the long-tail distribution in order to obtain a high-order hyperexponential approximation than it is to apply our algorithm directly to the empirical distribution generated from the data. Finally, we state our conclusions in Section 7.

2 The Robustness of Performance Models

Since we intend to approximate component distributions in performance models by other distributions, it is important that the performance models be robust to such changes. As a specific example, we will consider approximating long-tail service-time distributions by hyperexponential distributions in the $GI/G/1$ queue. (The $GI/G/1$ queue is just one example; There are many possible applications of hyperexponential approximations besides the $GI/G/1$ queue). The $GI/G/1$ queue has a single server, unlimited waiting room and interarrival times and service times coming from independent sequences of independent and identically distributed random variables with general distributions. If we approximate the given general interarrival-time and service-time distributions by other distributions, then we want descriptive performance measures such as the steady-state waiting-time distribution also to be approximately what it would be with the original interarrival-time and service-time distributions. Fortunately, such robustness, stability or continuity properties have been established for performance models.

Even though robustness results have been established, care is needed because the robustness results do not hold unconditionally. The robustness depends upon what we mean by "close" and upon regularity conditions. We say that a sequence of random variables $\{X_n : n \geq 1\}$ converges in distribution to a random variable X, and write $X_n \Rightarrow X$, if $Ef(X_n) \rightarrow Ef(X)$ as $n \rightarrow \infty$ for all bounded continuous real-valued functions f. If F_n and F are the cdf's of X_n and X, i.e., $F_n(t) = P(X_n \leq t)$, then $X_n \Rightarrow X$ is equivalent to convergence of cdf's in the form $F_n(t) \rightarrow F(t)$ as $n \rightarrow \infty$ for all points t that are continuity points of the limiting cdf F, which we denote by $F_n \Rightarrow F$.

With this background, we can state a robustness theorem for the $GI/G/1$ queue due to Borovkov [4] p. 118. A random variable is said to be proper if it is finite with probability one.

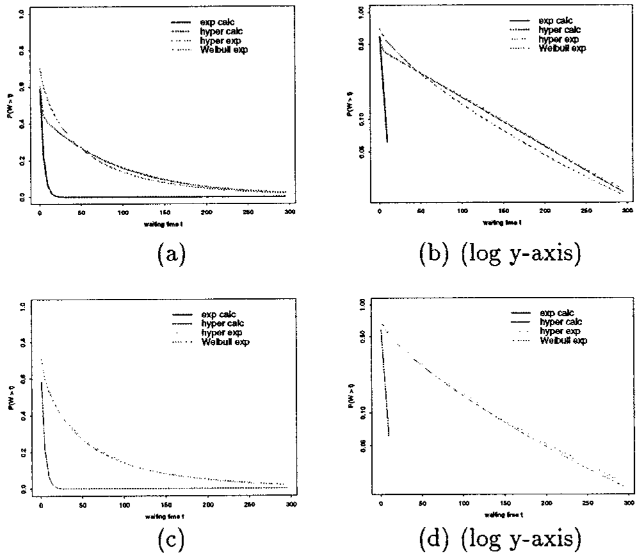
Theorem 2.1. (Borovkov). *Consider a sequence of $GI/G/1$ queueing models indexed by n with interarrival times, service times and steady-state waiting-time distributed as $U^{(n)}$, $V^{(n)}$ and $W^{(n)}$, respectively. Consider a prospective limiting $GI/G/1$ model with corresponding random variables U, V and W. If $EV^{(n)} < EU^{(n)}$ for all n, $EV < EU$, $U_n \Rightarrow U$, $V_n \Rightarrow V$ and $EV_n \rightarrow EV$ as $n \rightarrow \infty$, then $W^{(n)}, n \geq 1$, and W are proper random variables and $W_n \Rightarrow W$ as $n \rightarrow \infty$.*

The condition $EV^{(n)} < EU^{(n)}$ in Theorem 2.1 is needed in order to ensure that the n^{th} model is stable, i.e., that a proper steady-state waiting-time $W^{(n)}$ exists. An important point in Theorem 2.1 is that we also need to assume that the limiting system is stable ($EV < EU$), that the mean service times converge ($EV_n \rightarrow EV$), and that the limiting mean is necessarily finite ($EV < \infty$ since $EV < EU$). We need to assume that $EV_n \rightarrow EV$ as $n \rightarrow \infty$, because convergence in distribution does not imply convergence of moments. As a secondary point, note that there is no requirement that the mean interarrival times $EU^{(n)}$ and EU be finite or that $EU^{(n)} \rightarrow EU$ as $n \rightarrow \infty$.

To illustrate how we can apply Theorem 2.1, suppose that a $GI/G/1$ queueing system of interest has a generic service time V with a Pareto distribution as in (1.5). In the next section we will show that, without imposing any moment conditions, we can approximate the Pareto distribution of V arbitrarily closely by a hyperexponential distribution as in (1.7); i.e., for each n we can let $V^{(n)}$ have a hyperexponential distribution (where the number of component exponentials depends on n) and have $V^{(n)} \Rightarrow V$ as $n \rightarrow \infty$.

We would like to deduce that $W^{(n)} \Rightarrow W$ for the waiting-times in the associated $GI/G/1$ models. (Assume that the interarrival-time distribution is fixed.) However, we cannot draw this conclusion without the extra conditions in Theorem 2.1. The crucial extra condition is that $EV < \infty$; for the $GI/G/1$ application we must require that the Pareto distribution have a finite mean. If $EV = \infty$, then the approximation procedure will fail, but if $EV < \infty$, then it will work. It turns out that we can choose the approximating distributions so that $EV^{(n)} \rightarrow EV$ as $n \rightarrow \infty$, and we need to do so, but we also need to require that $EV < \infty$ and $EV < EU$ as well. However, with such extra conditions, approximating component distributions can achieve the desired result. The remaining questions are only the practical ones: How many exponentials are needed before the distribution of $V^{(n)}$ is suitably close to the distribution of V? And how do we actually find a good approximating distribution?

Example 2. To illustrate the robustness of the queueing model, we consider the Weibull distribution in Example 1 as a service-time distribution in the $M/G/1$ queue (having an exponential interarrival-time distribution). We let the arrival rate (and thus the traffic intensity) be 0.75. We focus on the steady-state waiting-time ccdf $P(W > t)$. In addition to the three-moment H_2 fit and the H_{20} fit by our algorithm in Section 4, we consider a simple exponential fit obtained by matching only the mean.



(a)

(b) (log y-axis)

(c)

(d) (log y-axis)

Figure 2: The steady-state $M/G/1$ waiting-time ccdf $P(W > t)$ with a Weibull(.3, 9.261) service-time distribution having mean = 1. the numerical results (calc) are for the model with the approximating hyperexponential and exponential service-time distributions. The simulations (exp) are for the model with the Weibull and the approximating hyperexponential distribution. Parts (a) and (b) contain the H_2 fit by matching the first three moments, while parts (c) and (d) contain the H_{20} fit by the algorithm in Section 4. Parts (b) and (d) are the same as parts (a) and (c), respectively, but with the y-axis in log scale.

We compare numerical results (calc) for the $M/H_2/1$ and $M/M/1$ models to simulations (exp) of the $M/H_2/1$ and $M/W/1$ models in (a) and (b) of Figure 2 (W stands for Weibull). In contrast, we compare numerical results for the $M/H_{20}/1$ and $M/M/1$ models to simulations of the $M/H_{20}/1$ and $M/W/1$ models in (c) and (d) of Figure 2. In all cases, the steady-state waiting-time ccdf is displayed, with the y-axis being in log scale in (b) and (d).

The $M/M/1$ model is appealing, because the steady-state waiting-time cdf for it is available in closed form (a simple exponential plus an atom at the origin), but it yields a remarkably poor approximation. Clearly the service-time distribution beyond its mean matters greatly. The $M/H_2/1$ numerical results could be obtained in several ways; we used numerical transform inversion [1]. The simulations were based on a time interval of 5.3×10^6, which corresponds to about 4×10^6 arrivals.

From (a) and (b) of Figure 2, we see that the $M/H_2/1$ approximation for the waiting-time ccdf is much better than the H_2 approximation for the W service-time distribution directly. This reflects the extensive experience showing that approximations based on two moments of the interarrival-time and service-time distributions can be quite effective. However, even though the $M/H_2/1$ approximation might be good enough for some engineering applications, the $M/H_{20}/1$ approximation in (c) and (d) is far better.

3 Complete Monotonicity

To have a good theoretical basis for approximating one distribution by another, it is appropriate to consider what is possible. From this perspective, it is important to note that every hyperexponential distribution has a decreasing probability density functions (pdf) and possibly an atom at 0. Thus, hyperexponential distributions cannot capture departures from this structure, such as atoms away from 0 or a non-monotone pdf.

On the other hand, there is a large class of distributions (necessarily with monotone pdf's) which can be approximately arbitrarily closely by hyperexponentials. The nice class of probability distributions are those with completely monotone pdf's. A probability density function (pdf) f is said to be *completely monotone* if all derivatives of f exist and

$$(-1)^n f^{(n)}(t) \geq 0 \quad \text{for all } t > 0 \text{ and } n \geq 1 \; ; \qquad (3.1)$$

see p. 439 of [8]. the link between completely monotone pdf's and mixtures of exponential pdf's is provided by Bernstein's theorem [8].

Theorem 3.1. (Bernstein) *Every completely monotone pdf f is a mixture of exponential pdf's, i.e.,*

$$f(t) = \int_0^\infty \lambda e^{-\lambda t} dG(\lambda) \; , \quad t \geq 0 \; , \qquad (3.2)$$

for some proper cdf G.

We call G in (3.2) the *spectral* cdf. (Then the support of G is called the *spectrum*. The *support* of G is the set of all t for which $G(t+\epsilon) - G(t-\epsilon) > 0$ for all $\epsilon > 0$.) Of course, the spectral cdf G appearing in (3.2) is a general cdf; it need not have finite support. (A cdf G has finite support if it has a probability mass function attaching probabilities p_i to n points t_i with $p_1 + \ldots + p_n = 1$ for some n.) However, cdf's with finite support are dense in the family of all cdf's. Hence, Theorem 3.1 implies the following result.

Theorem 3.2. *If F is a cdf with a completely monotone pdf, then there are hyperexponential cdf's $F^{(n)}$, $n \geq 1$, i.e., cdf's of the form*

$$F^{(n)}(t) = \sum_{i=1}^{k_n} p_{ni}(1 - e^{-\lambda_{ni} t}) \; , \quad t \geq 0 \; , \qquad (3.3)$$

with $\lambda_{ni} \leq \infty$ and $p_{n1} + \ldots p_{nk_n} = 1$, such that $F^{(n)} \Rightarrow F$ as $n \to \infty$.

Theorems 3.1 and 3.2 are important for approximating long-tail distributions because many long-tail pdf's are completely monotone. For example, by differentiating (and using mathematical induction), it is easy to see that the pdf's of the Pareto distribution in (1.5) and the Weibull distribution with $a < 1$ in (1.6) are completely monotone. The gamma pdf with shape parameter less than 1 is also completely monotone. The Pareto mixture of exponentials (PME) distribution considered in [1] is also completely monotone, because it directly satisfies (3.2). The PME distribution is convenient because its Laplace transform is available.

4 The Recursive Fitting Procedure

In this section we specify the recursive procedure for fitting a hyperexponential (H_k) cdf H to a given cdf F on the nonnegative real line. We think of the original cdf as being a long-tail distribution such as Pareto or Weibull with exponent less than one. We think of the cdf F as having a monotone probability density function (pdf) f, but we do not require it. We discuss conditions under which the procedure should be effective in Section 5.

The H_k distribution has ccdf (1.7). Without loss of generality, let the exponential parameters λ_i be labeled so that $\lambda_1 < \ldots < \lambda_k$. Then the higher indexed components have tails which decay more rapidly. Our idea is to fit the H_k components recursively, starting with the pair (λ_1, p_1) and then proceeding to (λ_2, p_2) and so forth. If λ_2 is sufficiently greater than λ_1, then $\sum_{i=2}^{k} e^{-\lambda_i t}$ should be negligible compared to $p_1 e^{-\lambda_1 t}$ for t sufficiently large (in the tail). This should enable us to choose the pair (p_1, λ_1) without being concerned about the other H_k parameter values. We then subtract the component $p_1 e^{-\lambda_1 t}$ from both $H^c(t)$ and $F^c(t)$ and fit the second component to the remaining tail. If again λ_3 is sufficiently greater than λ_2, then $\sum_{i=3}^{k} e^{-\lambda_i t}$ should be negligible compared to $p_2 e^{-\lambda_2 t}$ for t sufficiently large, and we can fit the pair (λ_2, p_2) without being concerned about the other H_k parameters.

After deriving this recursive fitting procedure, we learned that the general recursive estimation procedure actually has a long history, being known as Prony's method (p. 114 [16]). In that context, we contribute by showing when the recursive fitting procedure should be effective (Sections 3 and 5 here) and by applying it to approximate long-tail distributions.

Here is the procedure: we first choose the number k of exponential components and k arguments where we will match quantiles: $0 < c_k < c_{k-1} < \ldots < c_1$. We assume that the ratios c_i/c_{i+1} are sufficiently large; e.g., we could have $c_i = c_1 10^{-(i-1)}$ for $2 \leq i \leq k$. Let b be such that $1 < b < c_i/c_{i+1}$ for all i; e.g., with $c_i = c_1 10^{-(i-1)}$ we could have $b = 2$.

We choose λ_1 and p_1 to match the ccdf $F^c(t)$ at the arguments c_1 and bc_1; i.e., we solve the two equations

$$p_1 e^{-\lambda_1 x c_1} = F^c(xc_1) \text{ for } x = 1 \text{ and } b$$

for p_1 and λ_1, assuming that $c_1, b, F^c(c_1)$ and $F^c(bc_1)$ are known, obtaining

$$\lambda_1 = \frac{1}{(b-1)c_1} \ln\left(\frac{F^c(c_1)}{F^c(bc_1)}\right) \text{ and } p_1 = F^c(c_1)e^{\lambda_1 c_1}.$$

With this procedure, we are assuming that λ_i will be sufficiently larger than λ_1 for all $i \geq 2$ that the final approximation will satisfy

$$\sum_{i=1}^{k} p_i e^{-\lambda_i t} \approx p_1 e^{-\lambda_1 t} \quad \text{for } \; t \geq c_1 \; .$$

We have no guarantee that this property will hold, but the accuracy can be checked when the fit is complete. (See Section 5 for further discussion.)

Next, for $2 \leq i \leq k$, let

$$F_i^c(xc_i) = F_{i-1}^c(xc_i) - \sum_{j=1}^{i-1} p_j e^{-\lambda_j x c_i} \text{ for } x = 1 \text{ and } b$$

where $F_1^c(t) = F^c(t)$. Then proceed as above, letting

$$p_i e^{-\lambda_i x c_i} = F_i^c(xc_i) \text{ for } x = 1 \text{ and } b$$

to obtain

$$\lambda_i = \frac{1}{(b-1)c_i} \ln\left(\frac{F_i^c(c_i)}{F_i^c(bc_i)}\right) \text{ and } p_i = F_i^c(c_i)e^{\lambda_i c_i}$$

for $2 \leq i \leq k-1$. Finally, for the last parameter pair (λ_k, p_k), we require that

$$p_k = 1 - \sum_{j=1}^{k-1} p_j \text{ and } p_k e^{-\lambda_k c_k} = F_k^c(c_k),$$

where $F_k^c(c_k)$ is defined, so that

$$\lambda_k = \frac{1}{c_k} \ln(p_k / F_k^c(c_k)) \; .$$

Assuming that we obtain probability weights ($p_i > 0$ for all i), and that the parameters λ_i are well separated, we should obtain a good fit. Assuming that we obtain probability weights, the procedure produces an H_k ccdf H^c that is larger than the original ccdf F^c at the matching points, i.e.,

$$H^c(xc_i) > F^c(xc_i), \; 1 \leq i < k, \; \text{for } x = 1 \text{ and } b.$$

However, if F^c is a long-tail distribution, then there will be a t_0 such that

$$F^c(t) \geq H^c(t) \quad \text{for all } \; t \geq t_0 \; .$$

Hence, it is important to choose c_1 sufficiently large that t_0 is beyond the region of interest.

Our implementation of the algorithm in software allows the user to proceed interactively, choosing new parameter settings as desired, after looking at tables and graphs of the results. The standard approach is to specify k, c_1, c_k, and b. Then the algorithm chooses the remaining c_i such that the ratio of c_i/c_{i+1} is constant and proceeds with the fitting procedure. An available alternative is to specify one point at a time, start with the pair (c_i, b_i), inspect the preliminary result, and continue by choosing the next pair (c_{i+1}, b_{i+1}).

When we are done, we calculate several moments of the H_k distribution and compare them to the moments of F if they are available. As numerical measures of achieved fitting accuracy, we compute the absolute and relative errors of the ccdf and cdf. For both, the cdf and the ccdf, the absolute error is

$$AE(F,t) = |H^c(t) - F^c(t)| = |H(t) - F(t)|\,.$$

A relative error for both the cdf and ccdf is

$$RE(F,t) = \frac{|H^c(t) - F^c(t)|}{\min\{F(t), F^c(t)\}}.$$

We graphically display these errors as functions of t over any requested interval (l, u). We calculate the curves by considering points whose logarithms are evenly spaced over (l, u).

To illustrate, we display the absolute and relative errors of the H_2 and the H_{20} fits to the Weibull distribution in Example 1 in Figure 3. It turns out that the H_{20} fit was done with $c_k = 10^{-7}$ and $c_1 = 9 \times 10^4$. Since c_1 is not large, there are somewhat large relative errors for the H_{20} cdf in the region $10^2 - 10^5$. However, the ccdf values in this region are very small, e.g., $F^c(10^2) = 4.25e - 4$, $F^c(10^3) = 1.88e - 7$, $F^c(10^4) = 3.79e - 14$, and $F^c(10^5) = 1.67e - 27$.

5 When Should the Procedure Work?

In this section we discuss conditions under which the fitting procedure in Section 4 should be effective. In particular, we point out that the procedure is natural for distributions with *decreasing failure rate* (DFR). to see this, note that the fitting formula for λ_i can be rewritten as

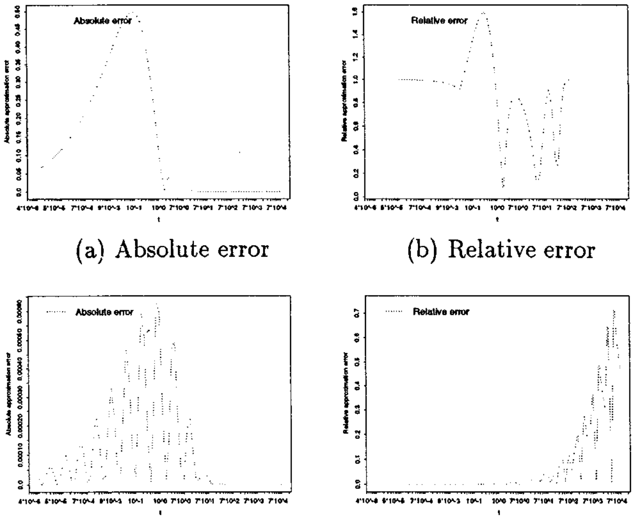
$$\lambda_i = -\frac{\ln(F_i^c(bc_i) - \ln(F_i^c(c_i)}{bc_i - c_i}\ . \tag{5.1}$$

As $b \to 1$, formula (5.1) approaches

$$\lambda_i = -\frac{d}{dt}\ln(F^c(t)|_{t=c_i} = \frac{f(c_i)}{F^c(c_i)} = r(c_i)\ , \tag{5.2}$$

which is the *hazard rate function* (or failure rate function) associated with the ccdf F^c evaluated at c_i [3].

The idea in the procedure of Section 4 is to have λ_i be significantly less than λ_{i+1} for all i. In order to have λ_i be less than λ_{i+1} for all i, it is natural to require that the ccdf $F^c(t)$ be DFR. This is equivalent to having $F^c(t)$ be *log-convex.* A sufficient condition for the ccdf $F^c(t)$ to be log-convex is for the pdf $f(t)$ to be log-convex (p. 73 [9]). Since mixtures of log-convex pdf's are log-convex (Theorem 5.4c, p. 66 [9]), all completely monotone pdf's are log-convex. Hence all completely monotone pdf's are DFR.



(a) Absolute error (b) Relative error

(c) Absolute error (d) Relative error

Figure 3: The relative and absolute errors of the H_2 and the H_{20} fits to the Weibull(0.3, 9.261) distribution from Example 1.

In summary, our algorithm is natural for completely monotone pdf's such as the Pareto and Weibull distributions (see Section 3) and, more generally, for DFR pdf's. However, by the same reasoning, our algorithm is inappropriate for increasing failure rate (IFR) distributions. For example, our algorithm does not work for the uniform distribution, i.e., when $F(t) = t/b, 0 \leq t \leq b$, and $F(t) = 1, t \geq b$, which clearly has a very short tail. Since many long-tail distributions are DFR, our algorithm has substantial applicability.

Even though many long-tail distributions are DFR, many others are not. Indeed, the long-tail property (1.1) is unaltered by changing the probability distribution on any initial interval $[0, t]$. Thus, the long-tail property does not nearly guarantee the DFR property.

6 Fitting to Data

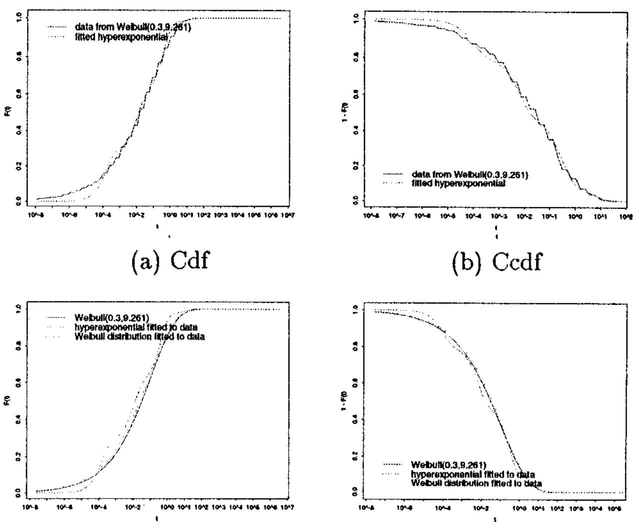
Besides using the fitting algorithm to fit a hyperexponential distribution to another distribution, we can also use the fitting algorithm to fit a hyperexponential distribution to data. In this case the empirical ccdf obtained from the data replaces the ccdf of the initial probability distribution in the algorithm.

However, we would suggest caution when applying our algorithm directly to data. Our experience is that it is usually much better to first fit a suitable long-tail probability distribution with only a few parameters to the data, and then afterwards apply our algorithm to fit a multi-parameter hyperexponential distribution to the long-tail distribution. By this two-step procedure, we usually are able to obtain a good multi-parameter hyperexponential fit to data.

We considered a simulation experiment in which we try to fit a probability distribution to a sample of 1000 points drawn from the Weibull(.3, 9.261) distribution considered in Example 1 having unit mean. Even though the sample size is not very large, it is large enough to obtain a good fit to the two-parameter Weibull distribution using the maximum likelihood estimator (p. 255 [10]). The Weibull parameters achieved from one sample were $c = 0.3016$ and $a = 9.369$ (yielding a mean of 0.96532). Since the estimated values of c and a are close to the original parameters, our algorithm applied to the fitted Weibull distribution can produce an excellent H_{20} approximation to the original Weibull distribution. For this experiment, the original Weibull distribution, the fitted Weibull distribution and the H_{20} fit to the fitted Weibull distribution are all very close, just as in Figure 1.

We also considered what happens when we apply our hyperexponential fitting algorithm directly to the data. Since the sample is not large, the range of the empirical ccdf is limited. Thus, it is not possible to directly apply our algorithm with many exponential terms. We show what happens with 4 exponentials. Figure 4 (a), (b) show how the fitted hyperexponential distribution matches the experimental cdf and ccdf. Figure 4 (c), (d) compare the fitted hyperexponential distribution to two Weibull distributions: the original Weibull distribution and the fitted Weibull distribution. Although the fits in Figure 4 look quite good, the pictures are deceptive, because the small and large values are not matched well. To illustrate, the moments are not matched well. For example, the third moments of the original Weibull cdf, the data, the H_4 fit and the Weibull fit were $4481, 1592, 145$, and 3820, respectively. The poor moments match can be explained in part by the fact that the sample moments of the data are not very close to the moments of the sampled distribution.

The experiment we have considered is somewhat biased, because we considered a hyperexponential fit to Weibull data. If we know in advance that the data is generated from the Weibull distribution, then using a statistical estimation procedure tailored to the Weibull distribution evidently should be good. It is less clear with an unknown data source. However, regardless of the data source, our fitting procedure is not designed to treat data. It does not address the statistical problems of the estimation. However, our procedure might well be applied effectively after some initial smoothing of the data, but that approach remains to be explored.



(a) Cdf

(b) Ccdf

(c) Cdf

(d) Ccdf

Figure 4: H_4 fit to the empirical cdf from a sample of size 1000 from a Weibull(.3, 9.261) distribution using $c_k = 0.0001$ and $c_1 = 5$.

7 Conclusions

In this paper we have developed an effective simple algorithm for approximating a large class of probability distributions with monotone densities by hyperexponential distributions (Section 4). We have found that the algorithm is effective for approximating Pareto and Weibull distributions (Sections 1, 2). We have shown that the algorithm should be effective for distributions with decreasing failure rate, and should not be used for distributions with increasing failure rate (Section 5). We have proved that, in principle, completely monotone pdf's (all of which have decreasing failure rate) can be approximated arbitrarily closely by hyperexponential pdf's, and that as a result (under extra regularity conditions) the associated waiting-time distribution in a $GI/G/1$ queue with a completely monotone service-time distribution can be approximated arbitrarily closely by the waiting-time distribution in the associated $GI/G/1$ queue with the approximating hyperexponential service-time distribution (Sections 2 and 3). Since many long-tail distributions are completely monotone, these results serve as a theoretical foundation for approximating long-tail distributions by hyperexponential distributions. Since phase-type probability distributions are dense in the family of all probability distributions, by the same reasoning, they are rich enough to approximate any distribution, if enough phases are allowed. We have pointed out that the EM algorithm is a candidate fitting algorithm for general phase-type distribution.

We believe that hyperexponential approximations of long-tail distributions can be useful, but they do not remove all difficulties. If a good fit is done, then the high variability of the long-tail distribution will be inherited by the approximating hyperexponential distribution. This high variability can make precise estimation by computer simulation difficult. Hyperexponential approximations also can make models of the superpositions of on-off sources more tractable, but since the state space of the Markovian environment process may be large, the approximating aggregate input process can still be difficult to analyze. However, we did see that the hyperexponential approximation makes it possible to calculate steady-state performance distributions in the $M/G/1$ queue with a long-tail service-time distribution by numerical transform inversion. The same technique applies to the more general $BMAP/G/1$ queue and other performance models.

We have emphasized that our fitting algorithm is intended to approximate one probability distribution by another, and not to fit a probability distribution directly to data (Section 6). In some circumstances our algorithm could be used to fit a hyperexponential distribution to an empirical distribution (histogram) obtained from data, but our algorithm is not designed for that purpose. Indeed, in simulation experiments with long-tail data, we found that much better fits are obtained by first fitting a long-tail distribution with very few parameters (e.g., 2) to the data and then applying our algorithm to obtain a hyperexponential distribution.

Finally, the algorithm presented here is only one of many possible fitting algorithms. We intend to compare alternative fitting algorithms in a future paper.

Acknowledgment

We thank our colleague William Turin for helpful discussions about the EM algorithm and the history of recursive estimation.

References

[1] J. Abate, G. L. Choudhury and W. Whitt, Waiting-time tail probabilities in queues with long-tail service-time distributions, *Queueing Systems* 16 (1994) 311–338.

[2] S. Asmussen, O. Nerman and M. Olsson, Fitting phase type distributions via the EM algorithm, *Scand. J. Statist.* (1996), to appear.

[3] R. E. Barlow and F. Proschan, *Statistical Theory of Reliability and Life Testing*, Holt, Rinehart and Winston, New York, 1975.

[4] A. A. Borovkov, *Stochastic Processes in Queueing Theory*, Springer-Verlag, New York, 1976.

[5] G. L. Choudhury, D. M. Lucantoni and W. Whitt, Squeezing the most out of ATM, *IEEE Trans. Commun.* 44 (1996) 203–217.

[6] M. E. Crovella and A. Bestavros. Self-similarity in World Wide Web traffic - evidence and possible causes. *Proceedings of Sigmetrics'96* (1996) 160-169.

[7] A. Feldmann. *On-line Call Admission for High-Speed Networks.* PhD thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 1995.

[8] W. Feller, *An Introduction to Probability Theory and Its Applications*, Wiley, New York, 1971.

[9] J. Keilson, *Markov Chain Models — Rarity and Exponentiality*, Springer-Verlag, New York, 1979.

[10] N. L. Johnson and S. Kotz, *Distributions in Statistics, Continuous Univariate Distributions*, Wiley, New York, 1970.

[11] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson. On the self-similar nature of Ethernet traffic. *IEEE/ACM Transactions on Networking* 2 (1994) 1–15.

[12] D. M. Lucantoni, The $BMAP/G/1$ queue: a tutorial, in *Models and Techniques for Performance Evaluation of Computer and Communication Systems*, L. Doniatiello and R. Nelson eds., Springer-Verlag, New York (1993) 330-358.

[13] M. Montgomery and G. de Veciana. On the relevance of time scales in performance oriented traffic characterizations. *IEEE INFOCOM'96* 513–520.

[14] M. F. Neuts, *Matrix-Geometric Solutions in Stochastic Models*, The Johns Hopkins University Press, Baltimore, 1981.

[15] V. Paxson and S. Floyd. Wide-area traffic: the failure of Poisson modeling. *IEEE/ACM Transactions on Networking* 3 (1995) 226–244.

[16] W. Turin, *Performance Analysis of Digital Transmission Systems*, Computer Science Press, New York, 1990.

[17] W. Turin, Fitting probabilistic automata via the EM algorithm, *Stochastic Models* 12 (1996) 405–424.

[18] W. Whitt, Approximating a point process by a renewal process, I: two basic methods, *Opns. Res.* 30 (1982) 125–147.

[19] W. Willinger, M. S. Taqqu, R. Sherman and D. V. Wilson, Self similarity through high variability: statistical analysis of Ethernet LAN traffic at the source level. *SIGCOMM'95* 100–113.