Imperial College of Science,
Technology and Medicine
(University of London)
Department of Computing

Finding Probability Distributions From Moments

by

Susanna W. M Au-Yeung

Submitted in partial fulfilment
of the requirements for the MSc
Degree in Computing Science of the
University of London and for the
Diploma of Imperial College of
Science, Technology and Medicine.

September 2003

Abstract

Using the moment sequence of a continuous probability function to regenerate the full distribution is a mathematical problem that has been investigated for many years. One method is to use a flexible distribution to approximate the densities by matching the moments of the two distributions. The results are of great interest, as they can be readily applied to response time analysis in concurrent systems, where obtaining moments is relatively easy but obtaining full distributions is hard. In this dissertation we look at the development of a tool that approximates densities when given the first four moments. The Generalized Lambda Distribution will be discussed in detail and we shall investigate the effectiveness of using it to approximate densities of well known probability distributions in addition to densities derived from response time analysis models.

Acknowledgements

I would like to thank everyone who supported me, especially my supervisor Dr William J. Knottenbelt for all his help and guidance and Nick Dingle for his time and patience.

Contents

1. Introduction

The problem of finding a probability distribution from its moment sequence has been long discussed in mathematics, and is known as “The Classical Moment Problem”. In this dissertation I’ll be looking at methods to develop a program that can plot density functions given the first four moments. I will be concentrating on looking at the use of the Generalized Lambda Distribution (GLD) as well as its effectiveness in approximating distributions and data..

In particular I shall be looking at the application of the GLD to approximate response time densities as first suggested in Guatema and van Gemund (2002). Response time densities and quantiles are important performance and quality of service metrics, but their analytical derivation is, in general, very expensive. It requires the solving of least 400 sets of linear equations, which usually increases to over 1000 sets compared to the 4 sets needed to obtain the first four moments.

An illustrative example of this is presented in Harrison and Knottenbelt (2002), which states that using a Laplace transform-based method, the probability distribution function of a response time Markov model with 1,211,354 states and 8.5 million transitions, was calculated in 26 minutes. This was using 32 slave PCs, each of which has a 1.4GHz AMD Athlon processor and 256MB RAM. The corresponding first three moments took just 5 minutes and 3 seconds to calculate using a single PC.

Therefore it is clear that if a method can be developed to approximate time response densities to a reasonable degree of accuracy, using only the moments of the distribution, it will be of great interest as it would help save much time and computing power.

The rest of this dissertation will proceed as follows:

I start with a brief introduction to the history and mathematical background of the classical moment problem. This is followed by a brief discussion of the possible routes by which a good approximation may be obtained.

There will then be an in depth discussion on the Generalized Lambda family of distributions, the range of shapes it offers and the method of fitting it to data and distributions using the first four moments.

Chapter 6 will highlight the implementation issues of using the GLD to produce a tool that approximates probability distribution functions and cumulative density functions of empirical data and well known distributions when given its first four moments.

I shall then be investigating the effectiveness of using the GLD and my tool in approximating some well known probability distributions as well as the quality of fit it provides in fitting data. In particular I shall be looking at the application of the GLD to response time densities in Markov and Semi-Markov stochastic models and comparing my results with those returned by the current tools.

Finally a check of how sensitive the GLD to the accuracy of moments entered will be investigated.

2. Background

2.1 The Classical Moment Problem

Every probability distribution has a moment sequence which is relatively straightforward to obtain, however the reverse problem of obtaining the probability distribution when given its moment sequence is a much harder proposition and is one that has been tackled by mathematicians for over a hundred years.

The term moment problem was first coined by the mathematician T. Stieltjes in 1894. The Stieltjes Moment Problem is as follows:

Given a certain sequence of numbers s_k (k = 0, 1, 2…), we seek a non-decreasing function F(x) where $x \geq 0$, such that

$$s_k = \int_0^{\infty} x^k \, dF(x) \quad \text{where } k = (0, 1, 2 \ldots)$$

By varying the interval in which F(x) is valid we obtain another 2 types of the classical moment problem:

The Hamburger Moment problem F(x) is defined in the interval $[-\infty, \infty]$.

The Hausdorff moment problem defines F(x) in the interval [0, 1].

The necessary and sufficient condition that must exist in order that the Hamburger Moment Problem is solvable for the moment sequence s_k where k = (0, 1, 2 …) is

$$D(s_0, s_1, \ldots, s_{2k}) > 0 \qquad \text{for } k = 0, 1, 2 \ldots$$

Where $D(s_0, s_1, \ldots, s_{2k})$ is the Hankel determinant of the moment sequence. The Hankel determinant (D) is defined as

$$D(a_0, a_1, \ldots, a_{2n}) = \begin{vmatrix} a_0 \ldots\ldots a_n \\ a_1 \ldots\ldots a_{k+1} \\ \ldots\ldots\ldots\ldots \\ a_n \ldots\ldots a_{2n} \end{vmatrix}$$

Similarly the solvability criteria for the Stieltjes Moment Problem is

$$D(s_0, s_1, \ldots, s_{2k}) \geq 0 \qquad \text{for } k = 0, 1, 2 \ldots$$
and $$D(s_1, s_2, \ldots, s_{2k+1}) \geq 0 \qquad \text{for } k = 0, 1, 2 \ldots$$

While for the Hausdorff Moment Problem we need

$$D(s_0, s_1, \ldots, s_{2k}) \geq 0 \qquad \text{for } k = 0, 1, 2 \ldots$$
$$D(s_1, s_2, \ldots, s_{2k+1}) \geq 0 \qquad \text{for } k = 0, 1, 2 \ldots$$
$$D(s_0\text{-}s_1, s_1\text{-}s_2, \ldots, s_{2k-1}\text{-}s_{2k}) \geq 0 \qquad \text{for } k = 0, 1, 2 \ldots$$
and $$D(s_1\text{-}s_2, s_2\text{-}s_3, \ldots, s_{2k}\text{-}s_{2k+1}) \geq 0 \qquad \text{for } k = 0, 1, 2 \ldots$$

However even if we find that the problem is solvable it is made even more complicated by the fact that in general, a distribution is not uniquely determined by its moments. Consider the following probability density function due to C.C Heyde

$$f_a(x) = \frac{1}{x\sqrt{2\pi}} e^{-1/2(\ln x)^2} (1 + a \sin(2\pi \ln(x)))$$

where $-1 \leq a \leq 1$.

The first six moments of $f_a(x)$ are $\{\sqrt{e}, e^2, e^{9/2}, e^8, e^{25/2}, e^{18}\}$ which we can see do not depend on a, in fact all the moments of $f_a(x)$ do not depend on the parameter a.

From Fig. 2.1 we see that for a = 0, a = -1/2 and a = -1/4 we have three very different probability density functions with the same moment sequence.

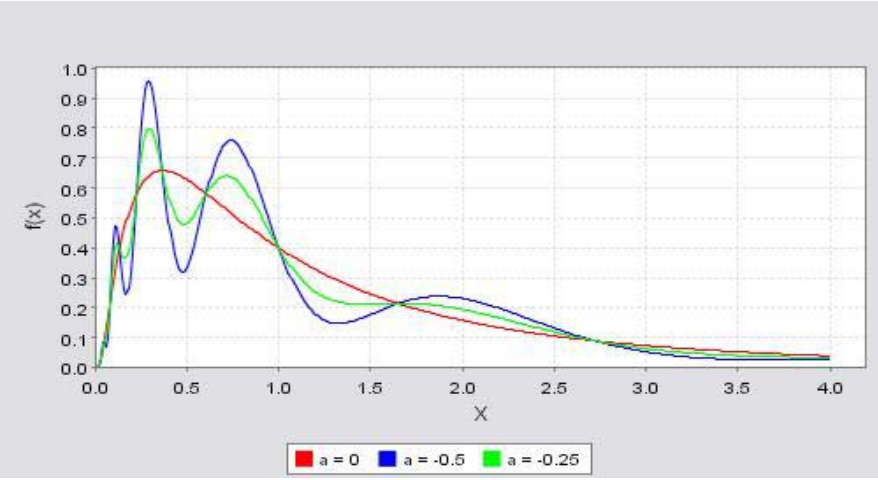


Fig. 2.1

To check whether an infinite sequence of moments determines a distribution uniquely we use Carleman's Theorem which states

A distribution on the interval $[-\infty, \infty]$ is determined uniquely by its moments μ_k if

$$\sum_{n=1}^{\infty} \frac{1}{\sqrt[2n]{\mu_{2n}}} = \infty$$

which means that if the series diverges the distribution is unique.

In general it is not feasible to obtain the infinite moment sequence and this is the case when looking at response time distributions. In our case we wish to address the related problem of reconstructing the probability density function from a finite sequence of moments. This problem arises often in both theoretical physics and quantum chemistry. We shall next discuss possible ways in which we can approach this problem.

2.2 Possible Methods of Finding a Solution

One approach is to use maximum entropy techniques to obtain an approximate probability function. This is a well documented area see for instance Jaynes (1994), however this method is mathematically complex and developing a tool that implements this method will be difficult.

Another possible method is to utilise and further refine an algorithm provided by Sándor Rácz (2002) in chapter 5 of his PhD dissertation, which describes a method to determine the upper and lower bounds of the set of cumulative distribution functions which match the given the first n moments. This algorithm is implemented by Rácz to form the WinMoments tool. However the derivation for the algorithm is not given and the theoretical background of the algorithm needs further development. In addition to this, the actual bounds obtained by the algorithm tend to be very large and so does not look like a promising starting point with which to obtain an actual distribution.

If we restrict the problem to unimodal probability density functions which the large majority of response time curves happen to be, we can use a great number of flexible distributions which can be straightforwardly applied to approximate the probability density function given its first few moments. I will now discuss three such possible distributions.

The **Pearson distributions** are a widely used family of distributions to approximate empirical data, with a wide diversity of distribution shapes. Each family in the Pearson system can be generated as a solution to the differential equation

$$\frac{df(x)}{dx} = \frac{(x - \varphi_3)f(x)}{\varphi_0 + \varphi_1 x + \varphi_2 x^2}$$

for the random variable x, with probability density function f(x).

Depending on the skew and kurtosis of the data you are approximating, you choose the appropriate Pearson family to use. However each family requires the solution of a different set of equations, which may be troublesome to implement, especially on the boundaries between families.

The **Johnson distribution** is a four parameter distribution based on the transformation of a standard normal variate. The Johnson distribution consists of 3 families and has been very successful in practise, with the 3 families make up differing regions in the skew-kurtosis plane. The principle drawback is that it is not very easy to determine the four parameters from the moments of the sample data. For more details on this and the Pearson distributions see Hahn and Shapiro (1967).

The **Generalized Lambda Distribution (GLD)** is a simple and flexible distribution that can assume a wide range of shapes and more importantly uses only one general formula, the parameters of which may be obtained when given the first four moments of the sample you wish to approximate.

The GLD's simplicity and versatility in fitting a broad range of curve shapes using only the first four moments, make the GLD an ideal candidate for further investigation into it's suitability in approximating response time distributions in concurrent systems.

3. Generalized Lambda Family of Distributions

3.1 Background

The Generalized Lambda Distribution (GLD) is a four-parameter generalization originally proposed by Ramberg and Schmeiser (1974) of the one-parameter Tukey-Lambda distribution introduced by Hastings et al in1947. Since then the flexibility of the GLD in assuming a wide variety of shapes has seen it being used extensively to fit and model a wide range of differing phenomena to continuous probability distributions, from applications in meteorology and modelling financial data, to Monte Carlo simulation studies.

The GLD is defined by an inverse distribution function or percentile (quantile) function. This is the function Q(u) where u takes values between 0 and 1, which gives us the value of x such that F(x) = u, where F(x) is the cumulative distribution function (cdf).

From this it is easy to derive the probability density function (pdf) for the GLD using differentiation by parts, however the cumulative distribution function needs to be calculated numerically.

The most popular method for estimating the GLD parameters is to match the first four moments of the empirical data to that of the GLD. The popularity of this method is partly due to the availability of extensive tables that provide parameter values for given values of skewness and kurtosis see Ramberg et al (1979) and Karian and Dudewicz (2000). In our case we will not be using tables to find the parameter values and will be calculating them directly.

However care must still be taken as different parameter values can return the same moments and hence the ensuing GLD may fail to properly represent the actual distribution of the data, this will be further discussed in section 6.4 with a good example of this is illustrated later in section 8.3.

3.2 Parameterizations of the GLD

In this section we will see three parameterizations of the Generalized Lambda Distribution. We shall define the original Tukey-Lambda distribution and two four-parameter generalizations, the traditional RS parameterization (Ramberg and Schmeiser 1974) and the FMKL parameterization (Freimer et al 1988).

The Tukey-Lambda distribution is defined by the percentile function Q(u).

$$Q(u) = \begin{cases} \dfrac{u^{\lambda} - (1-u)^{\lambda}}{\lambda} & , \lambda \neq 0 \\ \dfrac{\log(u)}{1-u} & , \lambda = 0 \end{cases} \qquad (1)$$

where $0 \leq u \leq 1$.

The original four-parameter generalization of (1) due to Ramberg and Schmesier (1974) which we shall call the RS parameterization is given by the percentile function Q(u).

$$Q(u) = \lambda_1 + \frac{u^{\lambda_3} - (1-u)^{\lambda_4}}{\lambda_2} \qquad (2)$$

where $0 \leq u \leq 1$.

The probability density function corresponding to (2) is given by:

$$f(x) = f(Q(u)) = \frac{\lambda_2}{\lambda_3 u^{\lambda_3 - 1} + \lambda_4 u^{\lambda_4 - 1}} \qquad (3)$$

where $0 \leq u \leq 1$.

Plotting the density for given $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ requires evaluating (2) and (3) for $0 \leq u \leq 1$ then plotting Q(u) against the corresponding f(Q(u)) for that value of u.

When using the method of moments to find the parameters of (2), in order for there to be a finite k^{th} moment we need $\lambda_3 > -1/k$ and $\lambda_4 > -1/k$. Since in order to find $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ we only need the first four moments, hence we need to impose the condition $\lambda_3 > -1/4$ and $\lambda_4 > -1/4$.

In addition to this Ramberg et al (1979) noted that there are certain combinations of λ_3 and λ_4 for which the distribution given by (2) is not a valid probability distribution. This undefined region is $1 + \lambda_3^2 < \lambda_4 < 1.8(\lambda_3^2 + 1)$ see Karian and Dudewicz (2000). The regions 1, 2, 3, 4, 5, 6 in Fig. 3.1 are the ones for which the distribution is valid is as follows:

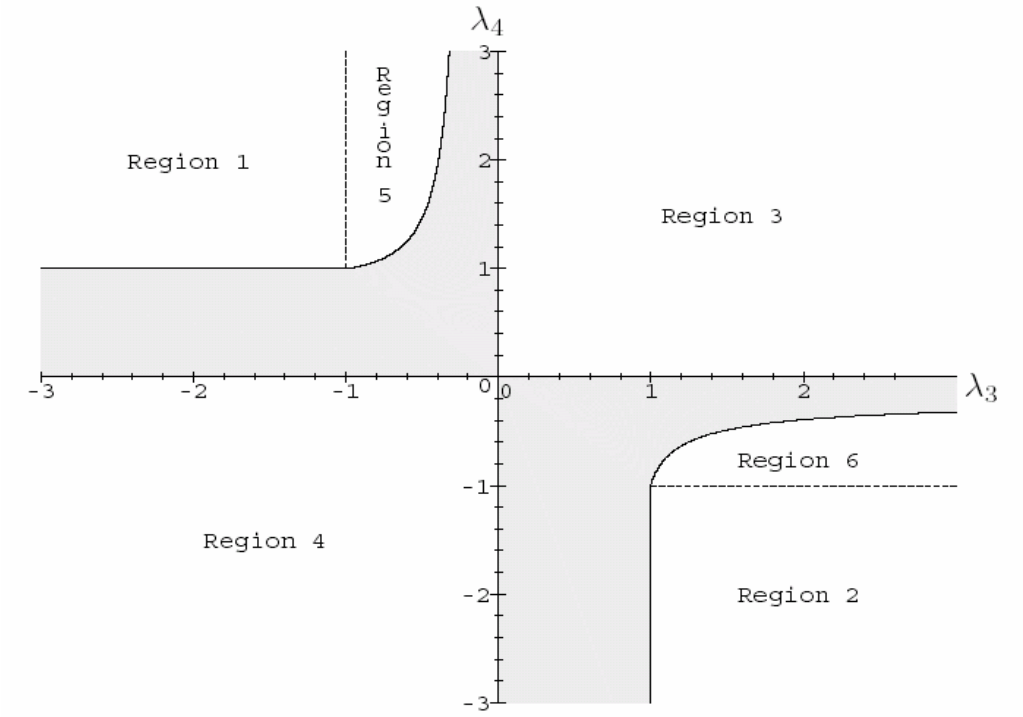


Fig. 3.1 Regions 1, 2, 3, 4, 5, 6 for which the RS parameterization is valid.

This limitation can be overcome in part by introducing a Generalized Beta Distribution (GBD) to partially cover the area not covered by the GLD. This addition of the GBD to the GLD together forms the Extended GLD (EGLD) described by Karian et al (1996) which together extends the valid region to $1 + \lambda_3^2 < \lambda_4 < 3 + 2\lambda_3^2$. See Karian and Dudewicz (2000) for an in depth study.

It is because of the limitations of the RS parameterizations and the extra complexity of additionally implementing the Generalized Beta Distribution to form the EGLD that I have decided to use the more convenient FMKL parameterization which is well defined over the entire λ_3, λ_4 plane. From now on when I refer to the GLD I am referring to the FMKL parameterization.

The FMKL parameterization developed by Freimer et al (1988) is given by Q(u).

$$Q(u) = \lambda_1 + \frac{1}{\lambda_2}\left[\frac{u^{\lambda_3} - 1}{\lambda_3} - \frac{(1-u)^{\lambda_4} - 1}{\lambda_4}\right] \qquad (4)$$

where $0 \le u \le 1$.

Using the relationships: $x = Q(u)$ and $F(x) = u$ and differentiating with respect to x, we get:

$$f(x) = f(Q(u)) = \frac{du}{dx} = \frac{du}{d(Q(u))} = \frac{1}{\frac{d(Q(u))}{du}} \qquad (5)$$

so by differentiating (4) and outing it into (5) we find f(x).

$$f(Q(u)) = \frac{\lambda_2}{u^{\lambda_3 - 1} + (1-u)^{\lambda_4 - 1}} \qquad (6)$$

Again if we are to use the method of moments to find the parameters in order to have finite moments we need $\lambda_3 > -1/4$ and $\lambda_4 > -1/4$.

3.3 Shape Characteristics of the FMKL Parameterization

The variety of shapes offered by this distribution includes unimodal, U-shaped, J-shaped, S-shaped and monotone probability distribution functions, which may be symmetric and asymmetric with smooth, abrupt, truncated, long, medium or short tails.

In order to classify the density shapes we need to know the role which each of the parameters λ_1, λ_2, λ_3, λ_4 play within the GLD:

λ_1 is the location parameter.
λ_2 determines the scale.
λ_3, λ_4 determine the shape characteristics.
For a symmetric distribution $\lambda_3 = \lambda_4$.

Freimer et al (1988) classify the shapes returned by (4) as follows:

Class I ($\lambda_3 < 1$, $\lambda_4 < 1$): Unimodal densities with continuous tails. This class can be subdivided with respect to the finite or infinite slopes of the densities at the end points. Class Ia (λ_3, $\lambda_4 \leq ½$), Class Ib ($½ < \lambda_3 < 1$, $\lambda_4 \leq ½$), and
Class Ic ($½ < \lambda_3 < 1$, $½ < \lambda_4 < 1$).

Class II ($\lambda_3 > 1$, $\lambda_4 < 1$): Monotone pdfs similar to those of the exponential or χ^2 distributions. The left tail is truncated.

Class III ($1 < \lambda_3 < 2$, $1 < \lambda_4 < 2$): U-shaped densities with both tails truncated.

Class IV ($\lambda_3 > 2$, $1 < \lambda_4 < 2$): Rarely occurring S-shaped pdfs with one mode and one antimode. Both tails are truncated.

Class V ($\lambda_3 > 2$, $\lambda_4 > 2$): Unimodal pdfs with both tails truncated.

Figures 3.2 to 3.8 show examples of each class of shapes.

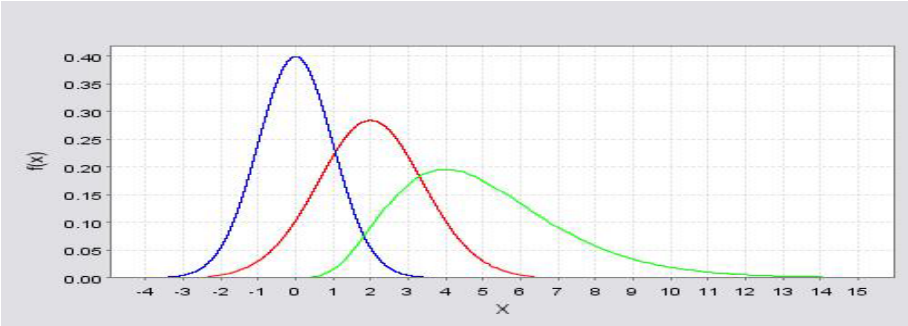


Fig. 3.2 Class Ia pdfs including the normal distribution.

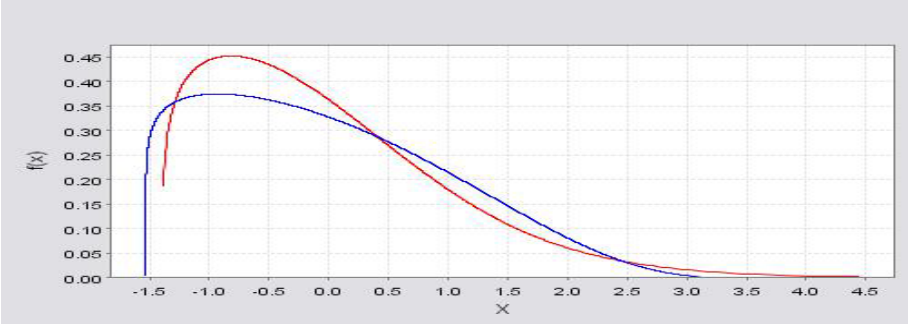


Fig. 3.3 Class Ib pdfs.

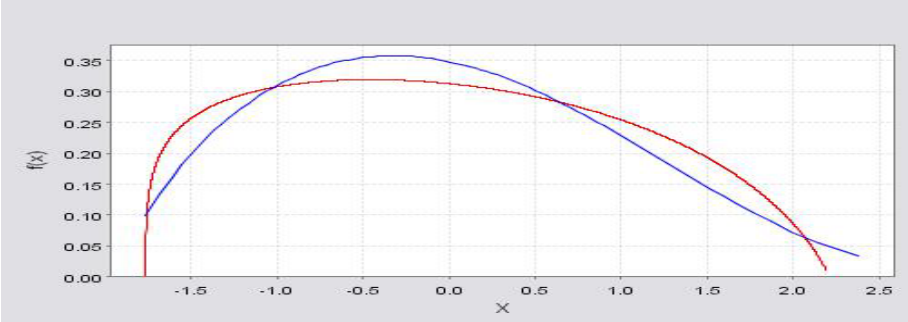


Fig. 3.4 Class Ic pdfs.

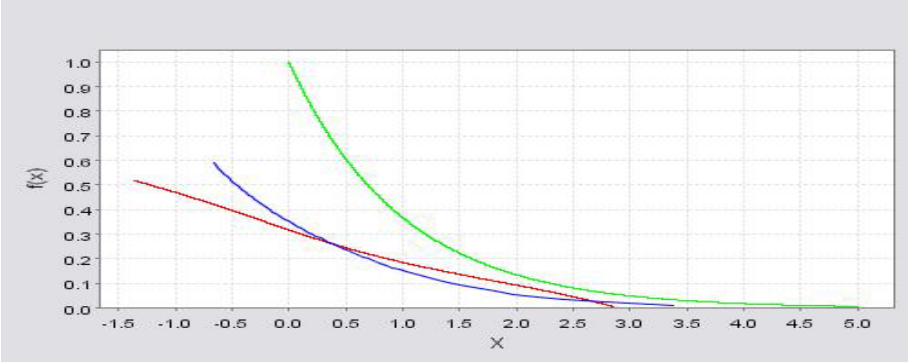


Fig. 3.5 Class II pdfs includes the exponential distribution.

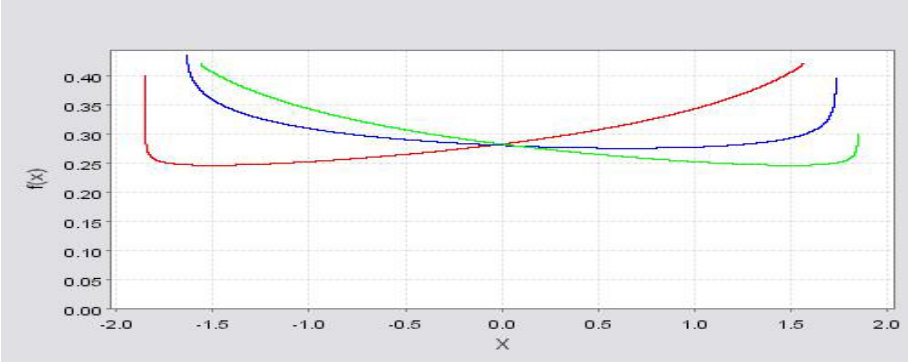


Fig. 3.6 Class III U-shaped pdfs.

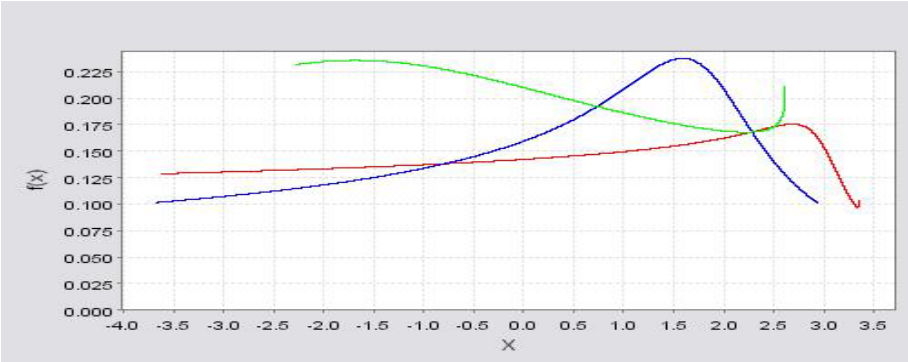


Fig. 3.7 Class IV S-shaped pdfs.

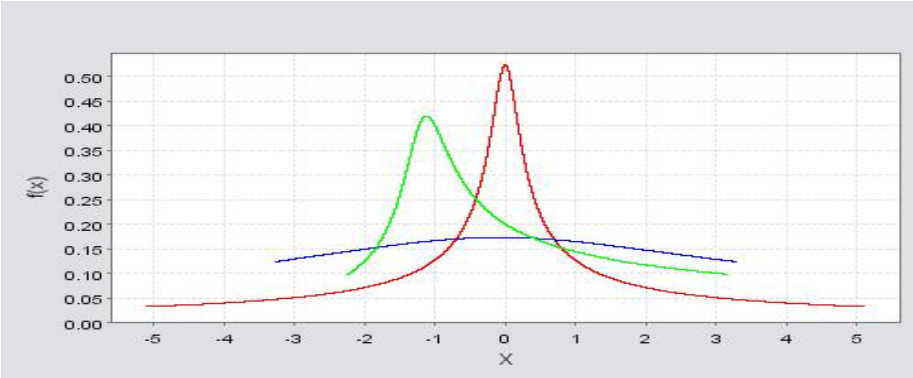


Fig. 3.8 Class V pdfs.

4. Using the GLD to fit Distributions and Data via moments

4.1 Background

Moments come in two forms. They are either raw (crude) moments or central moments.

The k^{th} raw moment of a probability density function f(x) of the random variable x is defined by:

$$E(X^k) \equiv \int_{-\infty}^{\infty} x^k f(x)\, dx \qquad \text{where } k \geq 1 \tag{7}$$

In particular the 1^{st} moment $E(X) = \mu$ is the mean or average of the function.

The k^{th} central moment is defined by:

$$E(X - \mu)^k \equiv \int_{-\infty}^{\infty} (x - \mu)^k f(x)\, dx \qquad \text{where } k > 1 \tag{8}$$

The central moments are expressed in terms of the raw moments by the following:

$$E(X - E(X))^k = \sum_{j=0}^{k} \binom{k}{j} (-X)^{k-j} E(X)^j \tag{9}$$

where $\binom{k}{j}$ are the binomial coefficients and $E(x)^0 = 1$.

When we use a percentile function Q(u), from equation (5) we have

$$\int_0^1 x^k f(x)\, dx = \int_0^1 Q(u)^k \frac{du}{d(Q(u))}\, d(Q(u))$$

Hence for the random variable X with the percentile function Q(u) the k^{th} raw moment is defined by

$$E(X^k) \equiv \int_0^1 Q(u)^k\, du \tag{10}$$

Mean, variance, skewness and kurtosis are values often used to describe the properties of a distribution.

In order to find the parameters of the GLD we match the mean μ, variance σ^2, skewness α^3, and kurtosis α^4 of the data sample to that of the GLD where:

$$\mu = E(X) \qquad (11) \qquad \sigma^2 = E(X - \mu)^2 \qquad (12)$$

$$\alpha^3 = \frac{1}{\sigma^3} E(X - \mu)^3 \qquad (13) \qquad \alpha^4 = \frac{1}{\sigma^4} E(X - \mu)^4 \qquad (14)$$

4.2 The Moments of the GLD

The moments of the FMKL parameterization of the GLD can be derived as follows

We can rewrite equation (4) as

$$F^{-1}(u) = \left[\lambda_1 - \frac{1}{\lambda_2\lambda_3} + \frac{1}{\lambda_2\lambda_4} + \frac{1}{\lambda_2} \left[\frac{u^{\lambda_3}}{\lambda_3} - \frac{(1-u)^{\lambda_4}}{\lambda_4} \right] \right] \tag{15}$$

$$= (b + a\, Q^*(u))$$

where

$$Q^*(u) = \frac{u^{\lambda_3}}{\lambda_3} - \frac{(1-u)^{\lambda_4}}{\lambda_4} \tag{16}$$

If X represents the random variable with percentile function Q(u) given in (4) and Y represents the random variable with percentile function Q*(u) given by (15) then from equation (14) we have

$$E(X) = \mu = (b + a\, EY)) \tag{17}$$

$$E(X - \mu)^k = a^k\, E(Y - E(Y))^k \tag{18}$$

Letting $v^k = E(Y)^k$ then from (10) we find that we need to calculate

$$v_k = \int_0^1 \frac{u^{\lambda_3}}{\lambda_3} - \frac{(1-u)^{\lambda_4}}{\lambda_4}\, du \tag{19}$$

Using binomial expansion we have

$$v_k = \int_0^1 \sum_{j=0}^{k} \binom{k}{j} (-1)^j \frac{u^{\lambda_3 (k-j)}}{\lambda_3^{k-j}} - \frac{(1-u)^{\lambda_4 j}}{\lambda_4^{j}}\, du$$

$$v_k = \sum_{j=0}^{k} \frac{(-1)^j}{\lambda_3^{k-j}\, \lambda_4^{j}} \binom{k}{j} \beta(\lambda_3(k-j) + 1, \lambda_4 j + 1) \tag{20}$$

where β is the beta function defined as

$$\beta(a,b) = \int_0^1 x^{a-1} (1-x)^{b-1}\, dx$$

The beta function will only converge if a and b are positive, hence in order for there to be finite moments we need:

$$\lambda_3(k-j) + 1 > 0 \text{ and } \lambda_4 j + 1 > 0$$

Since $0 \le j \le k$ and we only need the first four moments we need to impose the condition:

$$\min(\lambda^3,\lambda^4) > -1/4.$$

Using equation (20) we can obtain the following values

$$v_1 = \mu = \frac{1}{\lambda_3(\lambda_3 + 1)} - \frac{1}{\lambda_4(\lambda_4 + 1)}$$

$$v_2 = \frac{1}{\lambda_3^2(2\lambda_3 + 1)} + \frac{1}{\lambda_4^2(2\lambda_4 + 1)} - \frac{2}{\lambda_3\lambda_4}\beta(\lambda_3 + 1, \lambda_4 + 1)$$

$$v_3 = \frac{1}{\lambda_3^3(3\lambda_3 + 1)} - \frac{1}{\lambda_4^3(3\lambda_4 + 1)} - \frac{3}{\lambda_3^2\lambda_4}\beta(2\lambda_3 + 1, \lambda_4 + 1) + \frac{3}{\lambda_3\lambda_4^2}\beta(\lambda_3 + 1, 2\lambda_4 + 1)$$

$$v_4 = \frac{1}{\lambda_3^4(4\lambda_3 + 1)} + \frac{1}{\lambda_4^4(4\lambda_4 + 1)} + \frac{6}{\lambda_3^2\lambda_4^2}\beta(2\lambda_3 + 1, 2\lambda_4 + 1) - \frac{4}{\lambda_3^3\lambda_4}\beta(3\lambda_3 + 1, \lambda_4 + 1) - \frac{4}{\lambda_3\lambda_4^3}\beta(\lambda_3 + 1, 3\lambda_4 + 1)$$

From the above results and putting them into (18), we get the first four central moments of the FMKL parameterization of the GLD

$$E(X - \mu)^2 = \frac{1}{\lambda_2^2}(v_2 - v_1^2) \tag{21}$$

$$E(X - \mu)^3 = \frac{1}{\lambda_2^3}(v_3 - 3v_1v_2 + 2v_1^3) \tag{22}$$

$$E(X - \mu)^4 = \frac{1}{\lambda_2^4}(v_4 - 4v_1v_3 + 6v_1^2v_2 - 3v_1^4) \tag{23}$$

So from equations (13) and (14) we get the skewness and kurtosis of the GLD to be

$$\alpha_3 = \frac{v_3 - 3v_1v_2 + 2v_1^3}{(v_2 - v_1^2)^{3/2}} \tag{24}$$

$$\alpha_4 = \frac{v_4 - 4v_1v_3 + 6v_1^2v_2 - 3v_1^4}{(v_2 - v_1^2)^4} \tag{25}$$

So if we are given the mean μ^*, variance σ^{*2}, skewness α^*_3, and kurtosis α^*_4 of the sample data we kind find λ_3 and λ_4 of the GLD by solving

$$\alpha_3 = \alpha^*_3 \text{ and } \alpha_4 = \alpha^*_4 \tag{26}$$

Once we have found λ_3 and λ_4 we can find λ_2 from (25) and then λ_1 from (17) and using $a = 1/\lambda_2$, $b = \lambda_1 - 1/\lambda_2(1/\lambda_3 - 1/\lambda_4)$ we get

$$\lambda_2 = \frac{\sqrt{v_2 - v_1}^2}{\sigma^*} \tag{27}$$

$$\lambda_1 = \mu^* + \frac{1}{\lambda_2}\left[\frac{1}{\lambda_3 + 1} + \frac{1}{\lambda_4 + 1}\right] \tag{28}$$

Unfortunately exact solutions to (26) do not exist and we need to use numerical methods to obtain approximate solutions. This involves using optimization techniques to find λ_3 and λ_4 such that

$$(\alpha^*_3 - \alpha_3)^2 + (\alpha^*_4 - \alpha_4)^2 < \varepsilon \tag{29}$$

where ε is a positive number representing the accuracy to which we find the solutions.

5. Multi-dimensional Optimizing Techniques

5.1 Nelder-Mead Simplex Procedure

The Nelder-Mead simplex procedure due to Nelder and Mead (1965) provides an efficient way of finding a good approximation to a multi-dimensional function's optimum starting from any point, using only function evaluations.

A simplex is a geometrical figure which in n-dimensions consists of n+1 vertices. In two dimensions a simplex is a triangle, while in three dimensions it is a tetrahedron.

The procedure starts by picking n+1 points in the search space to define the initial simplex Fig. 5.1. The function to be minimized (in our case equation (29)) is evaluated at each of the vertices.

The procedure now takes a series of steps, starting with reflecting the worst (largest function value) vertex about the opposite face of the simplex to a lower vertex so that the volume of the simplex is preserved Fig. 5.2 If the new vertex now returns the best value the simplex is expanded in the direction of the new vertex Fig. 5.3 so that it can take larger steps. However if this new vertex still returns the worst value, the simplex is contracted along one-dimension away from the high point Fig.5.4.When the simplex is near a minimum the simplex is contracted in all directions towards the low point Fig 5.5. The procedure terminates when the function evaluation at each vertex is lower than a specified tolerance.

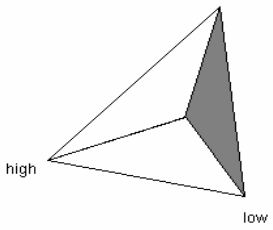


Fig. 5.1 Simplex at the start.

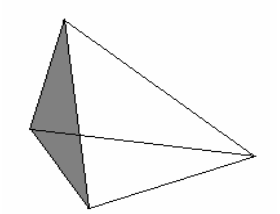


Fig. 5.2 Reflection.

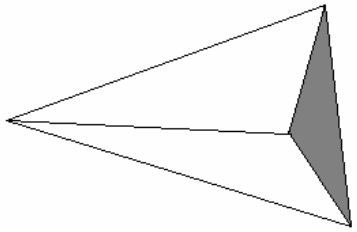


Fig. 5.3 Expansion

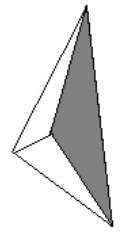


Fig. 5.4 Contraction in one-dimension

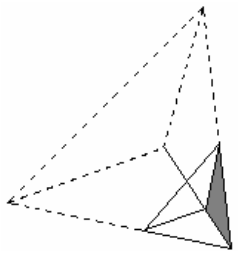


Fig. 5.5 Multiple contraction.

5.2 Powell's Method

The Nelder-Mead simplex procedure does not converge very well in the vicinity of the optimum, thus it is desirable to use the simplex algorithm to find an approximation to the optimum before using more precise methods to find the optimum to the desired level of accuracy. One such method is Powell's method.

Powell's method discovered by Powell (1962) is a hill climbing algorithm which finds an optimum in one direction before moving in the direction perpendicular to it in order to find an improvement, see Press et al (1992) for the mathematical background.

Besset (2001) recommends that Powell's algorithm be used only in the vicinity of the optimum and gives the following steps for the algorithm:

1. Let $\mathbf{x}_0$ be the n-dimensional vector representing the best point so far, and initialize a series of vectors $\mathbf{v}_1,\ldots,\mathbf{v}_n$ forming the system of reference, the components of the vector $\mathbf{v}_k$ are all zero except for the k^{th} component which is 1.

2. Set k =1.

3. Find the optimum of the goal function along the direction $\mathbf{v}_k$ starting from the point $\mathbf{x}_{k-1}$, where $\mathbf{x}_k$ is the position of that optimum.

4. Set k = k+1. If $k \leq n$, go back to step 3.

5. For k = 1,…,n-1, set $\mathbf{v}_k = \mathbf{v}_{k-1}$.

6. Set $\mathbf{v}_n = \dfrac{\mathbf{x}_k - \mathbf{x}_0}{|\mathbf{x}_n - \mathbf{x}_0|}$

7. Find the optimum of the goal function along the direction $\mathbf{v}_n$. Let $\mathbf{x}_{n-1}$ be the position of that optimum.

8. If $|\mathbf{x}_n - \mathbf{x}_0|$ is less than the desired precision, terminate.

9. Otherwise, set $\mathbf{x}_0 = \mathbf{x}_{n+1}$ and go back to step 1.

However both the Nelder-Mead simplex procedure and Powell's method have the same problem of terminating when reaching a local optimum. For this reason I next describe the Genetic algorithm introduced by John Holland in 1975. The random nature of this algorithm allows it to jump out of local minimums to search further for the absolute minimum.

5.3 The Genetic Algorithm

The Genetic algorithm is based on natural selection principles and tries to mimic the evolutionary process. The algorithm considers the elements of the search space as the chromosomes of individuals, so when optimizing a vector function, we take the genes to be the components of the vector. The function to be minimized is used as the measure of fitness of the individual to adapt to its environment. After each iteration the fittest survive and reproduce themselves with possible mutations and crossovers of the chromosomes also taken into account.

Mutation occurs during reproduction where one gene of a chromosome is altered. Crossover takes place when two chromosomes each break into two pieces before swapping one of the pieces with the other chromosome then reforming. The point at which the chromosomes break is called the crossover point. Where and when mutations and crossovers occur as well as which individuals survive are determined randomly.

For an in depth discussion on the implementation of the genetic algorithm, see Koza et al (1999)

6. Implementation

6.1 Programming issues

I have decided to use Java to implement the GLD, this is because Java is widely used, portable and has a large library of extensively tested code widely available.

The optimizing procedure I've used is the multi-variable optimizer as described in Besset (2000). It is a combination of the Genetic algorithm, the Nelder-Mead simplex procedure and Powell's method. The Genetic algorithm and Nelder-Mead simplex procedure are used to locate the general vicinity of the minimum, and then Powell's algorithm is used to locate the minimum to a greater level of accuracy.

I also need to implement an approximation to the Beta function which when can be defined as

$$\beta(a,b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$$

Where $\Gamma(t)$ is the Gamma function defined as

$$\Gamma(t) = \int_0^\infty e^{-x} x^{t-1} \, dx$$

I have used Lanczos approximation of the Gamma function (see Appendix A) which returns the value of the gamma function to a minimum accuracy smaller than 2×10^{-10}. In my program I have implemented $\ln \Gamma(t)$ so as to prevent overflow occurring.

As mentioned before, we cannot find the cumulative density function of the GLD exactly. Therefore we need to employ numerical methods to approximate the integral of the cumulative density function of the GLD.

The numerical method I have employed is the trapeze integration method which splits the interval on which the distribution is defined; we then use the resulting trapeziums to approximate the area under the curve as shown below

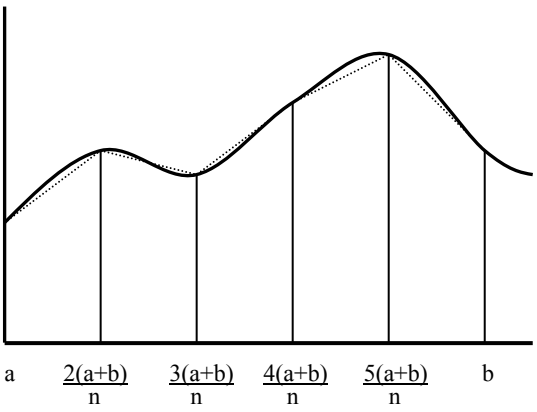


Fig. 6.1

Both raw and central moments are widely used, hence I have provided for the input of the first four moments to be either the mean, variance, skew and kurtosis or the first four raw moments.

In order to plot the GLD approximations to distributions and to see how they compare to the real thing, I used the open source graph plotting library JFreeChart (see www.object-refinery.com/jfreechart). When plotting the graphs, I have taken $0.0001 \leq u \leq 0.9999$ in equations (4) and (6) taking steps of 0.0001, giving me a mesh of 10,000 points with which to plot the values.

6.2 Accuracy

We need to use numerical methods to find the parameters of the GLD, hence we can only match the mean, variance, skewness and kurtosis of the GLD to that of the data to a certain level of accuracy. From (29) we know that

$$\max \quad | (\alpha^* - \alpha)^2 | < \varepsilon \qquad (30)$$

where ε is a positive number representing the tolerance to which we find the solutions and α represents either one of the skewness or kurtosis.

This means that the minimum accuracy to which can match the moments is given by $\sqrt{\varepsilon}$. I have allowed the user of my tool to specify the level of minimum accuracy (ie. the value of $\sqrt{\varepsilon}$) since for some sets of moments we are unable to find any solutions when the value of $\sqrt{\varepsilon}$ is set too small, however in general if you set $\sqrt{\varepsilon}$ too large you may return incorrect approximations. For the default setting I have set $\sqrt{\varepsilon} = 1.0 \times 10^{-4}$, in general parameters are returned to at least $\sqrt{\varepsilon} = 1.0 \times 10^{-8}$.

6.3 Testing

In order to see how effective the method of GLD's is in approximating distributions and data, I need to be able to quantify how accurate the method is. For the case when we are testing the GLD against well known distributions where there are given probability density functions it is easy to find the maximum difference between the two distributions by taking the absolute difference between the two at each of the 10,000 values of u (and corresponding x) then taking the maximum of these.

However when we test the GLD against data, the parametric nature of the GLD makes it hard to exactly quantify the exact difference between the GLD approximation and the data given. Because of this I have been unable to find an accurate way of quantifying the how well the GLD approximates data and can only provide graphs so that the user can visually determine how good the approximations are.

6.4 Extra Solutions

As previously mentioned, there at times where more than one set of parameter values exist for a set of moments. These different sets of parameters, for each set of parameters you will get a different approximation. For instance for the following values of mean, variance, skewness and kurtosis :

$$\mu = 0, \quad \sigma^2 = 1, \quad \alpha^3 = 0, \quad \alpha^4 = 3.$$

These are the values for the Standard Normal distribution, matching these to the GLD, you can get the following two sets of results:

λ_1 = -1.04895187 e-9, λ_2 = 1.4635577, λ_3 = 0.13490936, λ_4 = 0.13490936. (i)

λ_1 = -3.67987 e-11, λ_2 = 0.0803605, λ_3 = 5.202901559, λ_4 = 5.202901559. (ii)

Using set (i) to plot the GLD approximation to N(0,1) provides a good fit, however set (ii) would provide a very poor approximation. These two GLD plots can be seen in Fig. 6.2.

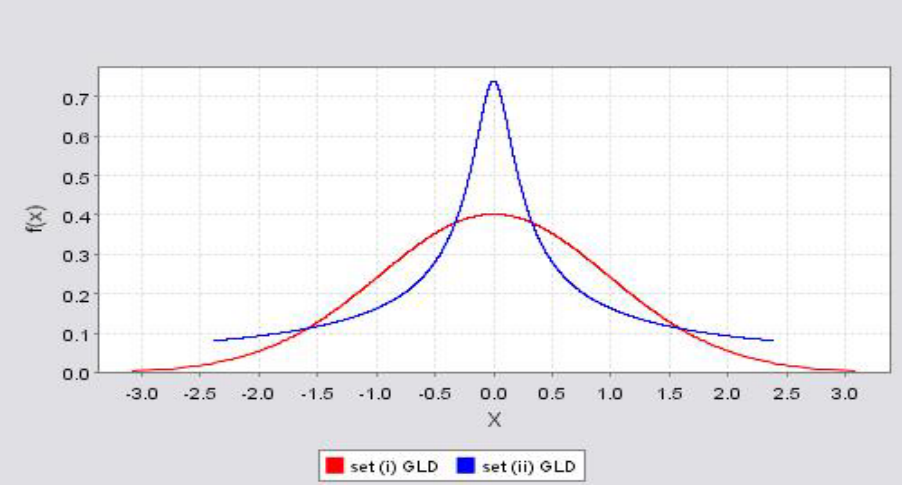


Fig. 6.2

Since my optimizer will simply return the first set of parameters it finds that matches the moments to the required level of accuracy, without regard as to whether or not there may be another solution, I need to find a way that ensures as much as possible that the most suitable set of parameters are used.

Unfortunately I have been unable to find a sure fire way of ensuring the best set of parameters are returned, however I have noticed that the unsuitable sets of parameters tend to be found to a higher level of accuracy, typically for values of $\varepsilon < 1.0 \times 10^{-10}$. It is for this reason that I have restricted my optimizer to return sets of parameters such that they are of the level of accuracy greater than that specified by the user, but less than 1.0×10^{-10} in order to reduce the likelihood of an unsuitable set of parameters being used. Through restricting the parameters you do get some degree of success, however this doesn't insure that the best set of parameters are always used, and you may still get some unsuitable approximations.

7. GLD Approximations to Some Well Known Distributions

7.1 The Normal Distribution

The normal distribution with mean μ, and variance σ², denoted by N(μ, σ²)has pdf

$$f(x) = \frac{1}{\sigma\sqrt{2}} \exp\left(\frac{-(x-\mu)^2}{2\sigma^2} \right) \quad -\infty < x < \infty$$

where $\alpha_3 = 0, \quad \alpha_4 = 3\sigma^4$

All normal distributions may be obtained form the Standard Normal distribution N(0, 1) using the Central Limit Theorem, therefore I have considered the GLD fit to N(0,1) where

$$\mu = 0, \quad \sigma^2 = 1, \quad \alpha^3 = 0, \quad \alpha^4 = 3.$$

My program returns GLD parameter values of

λ_1 = -1.04895187 e-9, λ_2 = 1.4635577, λ_3 = 0.13490936, λ_4 = 0.13490936.

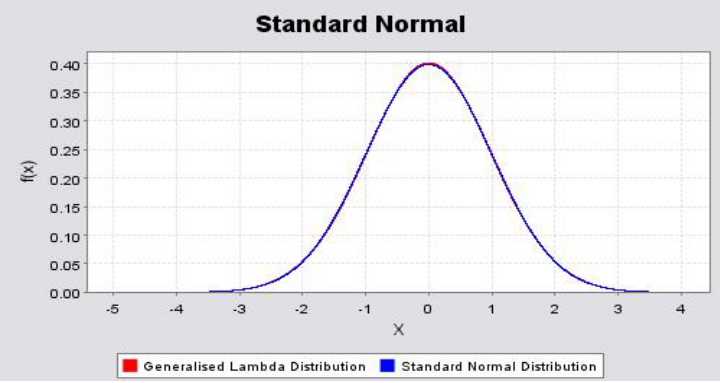


Fig. 7.1

Fig. 7.1 shows a plot of N(0,1) and the GLD approximation, as you can see the GLD provides a very good fit with the two distributions almost indistinguishable except near x = 0 where the GLD rises slightly higher.

If we denote the pdf of N(0,1) by f*(x) and that of the GLD by f(x) we get

$$\max |f(x) - f^*(x)| = 0.002813$$

So we can say that the GLD is accurate to within 0.002813.

Next we shall look at the cdf of N(0,1) and the corresponding GLD. Unfortunately we cannot find the cdf of the Normal distribution exactly however there are a number of good approximations, I have implemented the approximation due to Abramovitz and Stegun (1964) see Appendix A, which returns the cdf of N(0,1) , F*(x) to a minimum accuracy of 7.5 x 10^{-8}. The graph obtained is shown in Fig 7.2

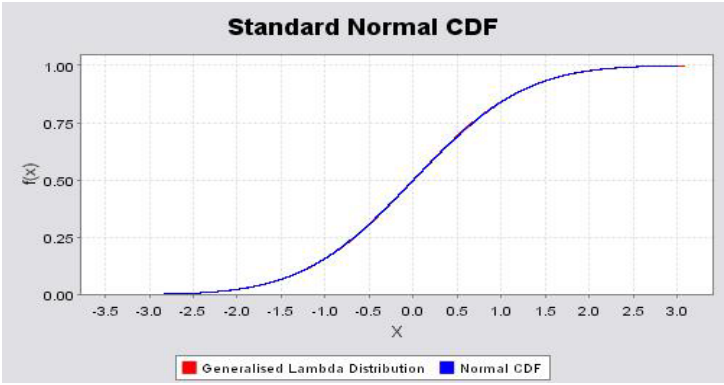


Fig. 7.2

From the graph above we can see that the GLD cdf provides a very good fit to that of the cdf of N(0,1), in fact the two curves are so close together you can hardly tell them apart. We also found

$$\text{Max } |F(x) - F^*(x)| = 0.00118$$

Which shows that the GLD provides an even better fit to the cdf than did the pdf, these results together show that the GLD provide a very good fit to the normal distribution.

7.2 The Uniform Distribution

The pdf of the uniform distribution on the interval [a,b], U(a,b) is as follows

$$f(x) = \frac{1}{a - b}$$

where

$$\mu = \frac{a + b}{2}, \quad \sigma^2 = \frac{(b - a)^2}{12}, \quad \alpha_3 = 0, \quad \alpha_4 = 1.8$$

Looking at U(0,1) for which $\mu = 0.5$ and $\sigma^2 = 0.08333333333$ we get GLD parameter values of

$$\lambda_1 = 0.5, \quad \lambda_2 = 2, \quad \lambda_3 = 1, \quad \lambda_4 = 1.$$

In this case the GLD will be a perfect fit since these values of $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ in (4) yield $Q(u) = u$ and thus $F(x) = u$ and $f(x) = 1$, which are the cdf and pdf of U(0,1). Fig. 7.3 shows the resulting graph for the cdf which you can see is exact.

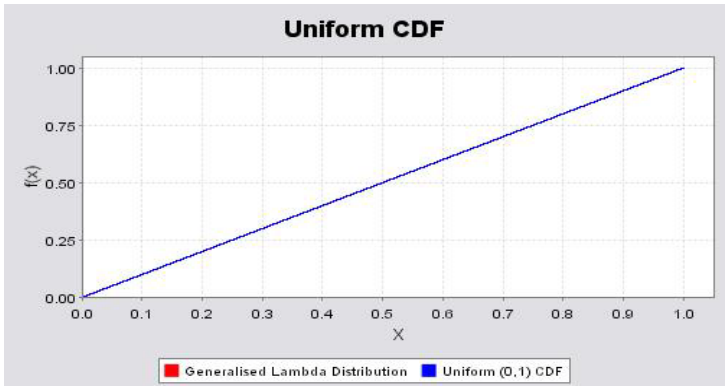


Fig. 7.3

7.3 The Exponential Distribution

The pdf of the Exponential distribution is given as

$$f(x) = \begin{cases} \frac{1}{\beta} e^{-x/\beta} & \text{for } x > 0 \\ 0 & \text{otherwise.} \end{cases}$$

where: $\mu = \beta, \quad \sigma^2 = \beta^2, \quad \alpha_3 = 2, \quad \alpha_4 = 6.$

The cdf is defined as: $F(x) = 1 - e^{-x/\beta}$

We can see that the values of α_3, α_4 remain unchanged for whatever value of β we use. Therefore we can use the values of λ_3 and λ_4 that we obtain for say $\beta = 1$ to all other exponential distributions.

Using $\beta = 1$, we have $\mu = 1$ and $\sigma^2 = 1$ which gives

$\lambda_1 = 0.154880287$, $\lambda_2 = 1.030033765$, $\lambda_3 = 6.784091898$, $\lambda_4 = 0.0010320893$.

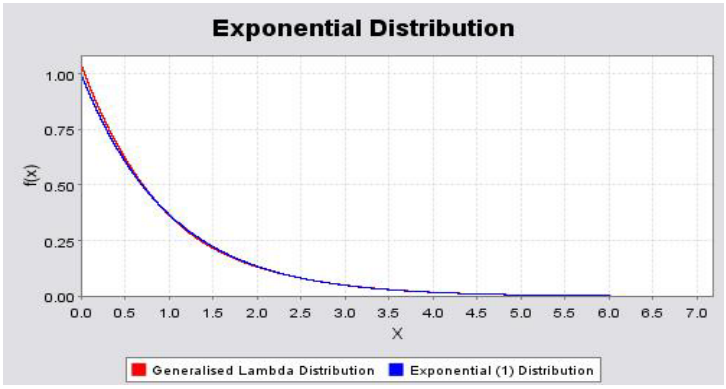


Fig. 7.4

We can see from Fig. 7.4 that the GLD is slightly higher initially but still provides a good fit, this is illustrated by: max $|f(x) - f^*(x)| = 0.038867$.

If we now look at how the cdfs compare, we find max $|F(x) - F^*(x)| = 0.0118$, and from Fig. 7.5 we see again that the GLD provides a good approximation.

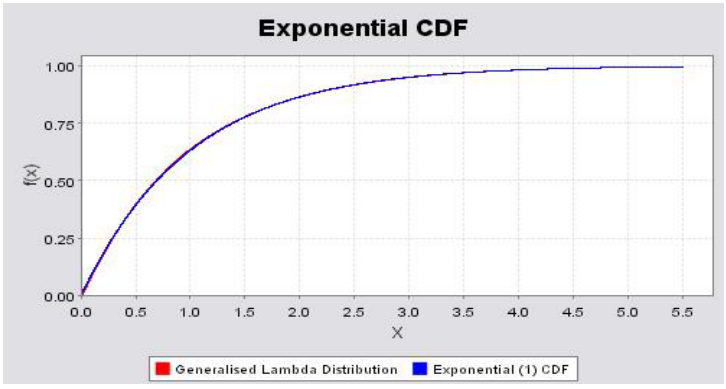


Fig. 7.5

7.4 The Gamma Distribution

The pdf of the Gamma distribution is defined as

$$f(x) = \frac{x^{\alpha-1}}{\beta^{\alpha}\Gamma(\alpha)} e^{-x/\beta} \qquad \text{where } x \geq 0.$$

and $\mu = \alpha\beta, \quad \sigma^2 = \alpha\beta^2, \quad \alpha_3 = \frac{2}{\sqrt{\alpha}}, \quad \alpha_4 = \frac{6}{\alpha}$.

As an example I have taken $\alpha = 5$ and $\beta = 3$ which gives

$$\mu = 15, \quad \sigma^2 = 45, \quad \alpha_3 = 0.89443, \quad \alpha_4 = 4.2$$

This returns GLD parameters of

$\lambda_1 = 13.6931635, \lambda_2 = 0.214337475, \lambda_3 = 0.40227341, \lambda_4 = 0.00681439986.$

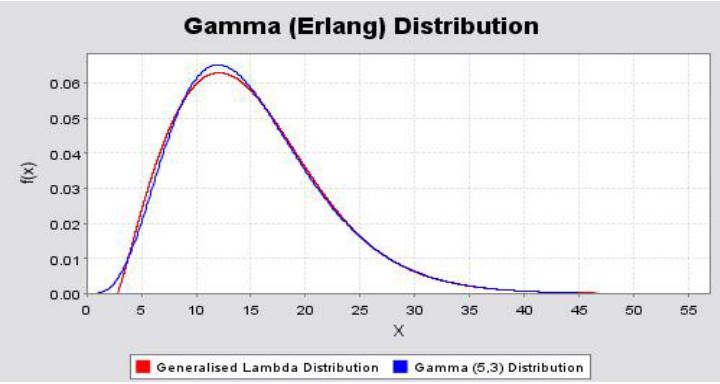


Fig. 7.6

We can see in Fig.7.6 that the GLD provides a reasonable fit, with the GLD slightly lower at the peak and starting with a larger initial x value. Calculating the difference we have: $\max |f(x) - f^*(x)| = 0.003748$.

In order to compare the cdfs I need to compute the incomplete gamma function, I have implemented this as shown in Appendix A. From Fig 7.7 we see what a good fit the GLD returns, giving:

$$\max |F(x) - F^*(x)| = 0.0064737.$$

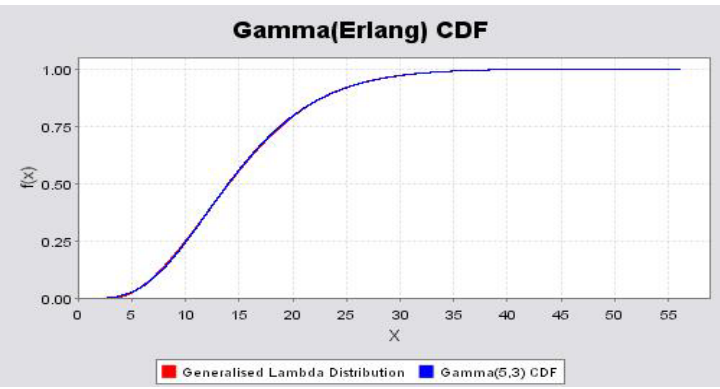


Fig. 7.7

8. GLD Approximations to Response Time Densities

I shall now be investigating how effective the GLD is in approximating the densities obtained from Markov, semi-Markov and GSPN response time models as well as the time taken to produce the approximations, which I will compare against the time taken to produce the corresponding exact densites for some of the models. In addition I shall be comparing some of my results against those obtained using the current best tool to approximate probability densities using the first four moments, known as the WinMoments tool. (See section 2.2)

8.1 A Semi-Markov Model

The first model I am going to approximate, called SM four. This is a semi-Markov Stochastic Petri net model with a "ring" of four places and four transitions see Fig.8.1. The transitions are a 4-stage rate 3 Erlang, a rate 5.7 Exponential, a Uniform (a = 1.3, b = 7.8) and a Deterministic 0.89, see Fig 8.1 below. For details on how the exact density is calculated see Bradley et al (2003). The time is the time taken for 20 tokens to move from place one, through places two and three to place zero.

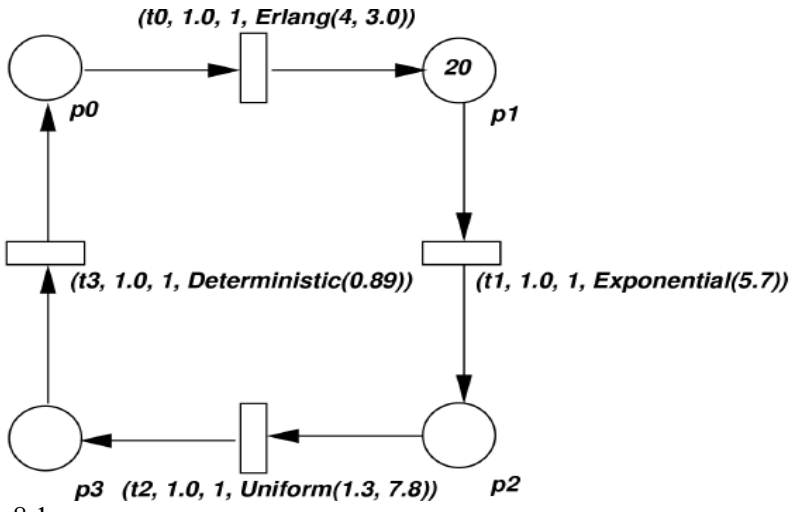


Fig 8.1

The first four raw moments of SM four are:

$$(252.391,\ \ 69314.9,\ \ 2.08091 \times 10^{7},\ \ 6.85231 \times 10^{9}).$$

Putting these raw moments in my tool I get the pdf shown in Fig. 8.2.

We can see that the GLD approximation has a lower peak with a larger initial x value, starting inside the actual density of the model. When we compare the two cdfs we obtain Fig. 8.3 from which we can see that the GLD provides a good approximation. In regards to the time taken to generate the approximation, using my program it took an average of 242.4 milliseconds to calculate the GLD parameters.

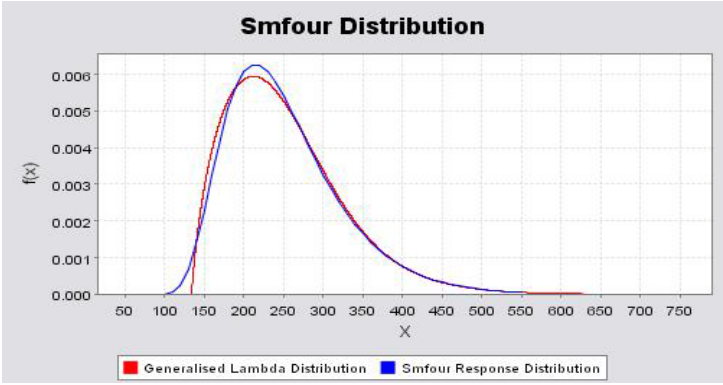


Fig. 8.2

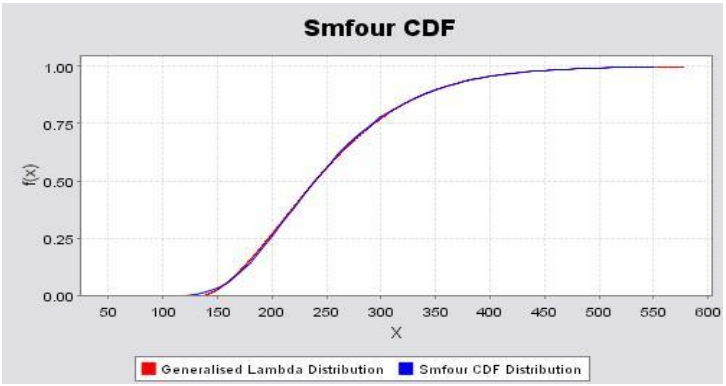


Fig. 8.3

The next Semi-Markov model we look at called Webserver, this is a model of a web-content–authoring system. Content authors place content updates in a buffer, which is written to the servers when at least one is operating and no readers are reading. Readers make requests for content to servers. Servers can fail and recover, time is the time taken for 24 read requests to have been submitted to the system.

The first four raw moments of Webserver are:

$$(36.148,\ 1433.45,\ 61753,\ 2.86678 \times 10^{6})$$

The GLD approximation is shown in Fig. 8.4. As you can see the GLD provides a very good approximation to the Webserver response time density, with the only main difference between the two being that the GLD approximation is slightly lower at the peak.

Since the pdfs are so similar, we would also expect the GLD to provide a very good approximation to the cdf of the Webserver model, the result is shown (Fig 8.5). As expected we see that the two cdfs are very close. Computing these approximations took on average 1.0774 seconds.

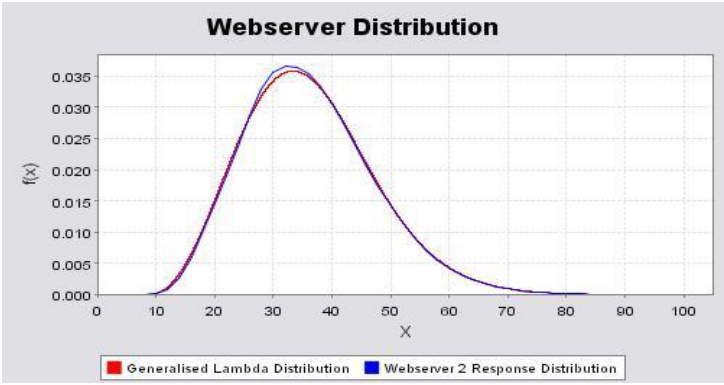


Fig. 8.4

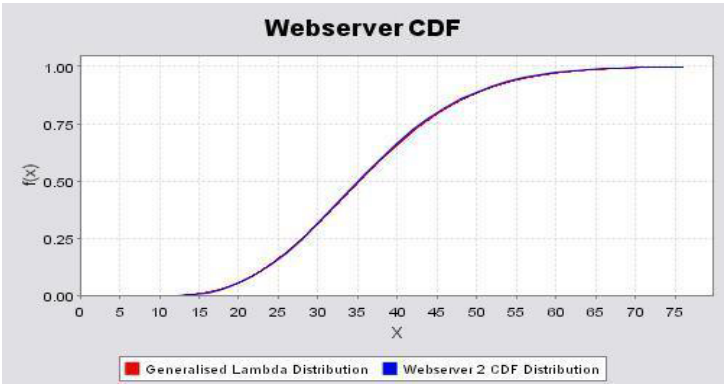


Fig. 8.5

If you compare the above GLD approximation to the Webserver cdf. to the bounds returned by the WinMoments tool, we see from the error bars in Fig 8.6, that the GLD method of approximation is well within the WinMoments bounds, and provides a better approximation to the actual cdf than the mid-point of the bounds.

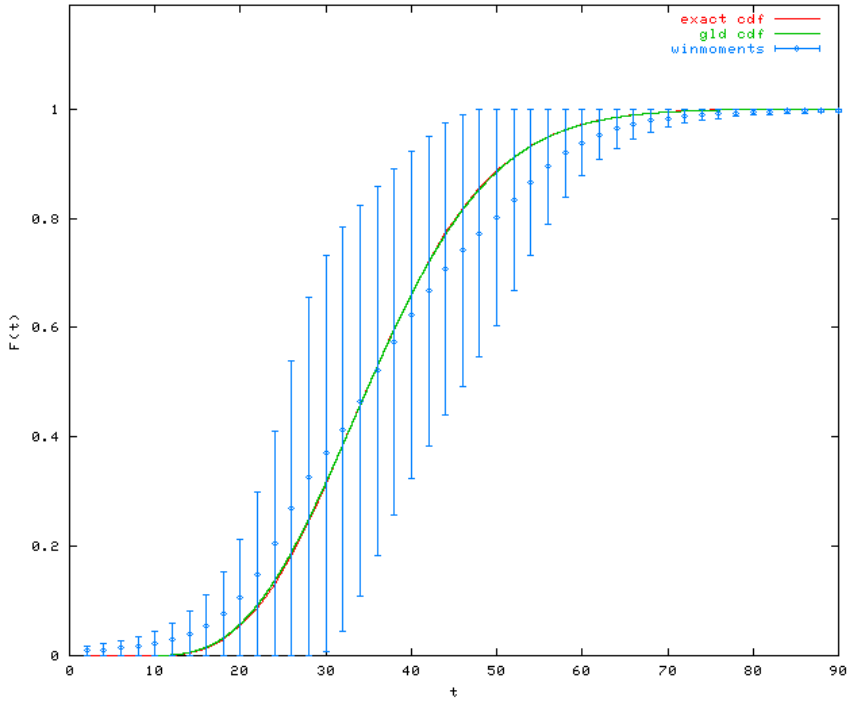


Fig 8.6 Comparison of the GLD approximation against the WinMoments tool

8.2 GSPN Models

I shall now approximate a two General Stochastic Petri Net models. I start with a 45 place GSPN model of a communication protocol, called Courier, representing the ISO Application, Session and Transport layers of a sliding-window communication protocol (details in Woodside and Li (1991). The response time is the time taken from the initialisation of a transport-layer send, to the arrival of the corresponding acknowledgement packet.

The first four raw moments of the Courier model are:

$$(26.528,\ \ 955.083,\ \ 43410.9,\ \ 2.39069\text{x } 10^{6})$$

This gives the GLD approximation shown in Fig 8.7.

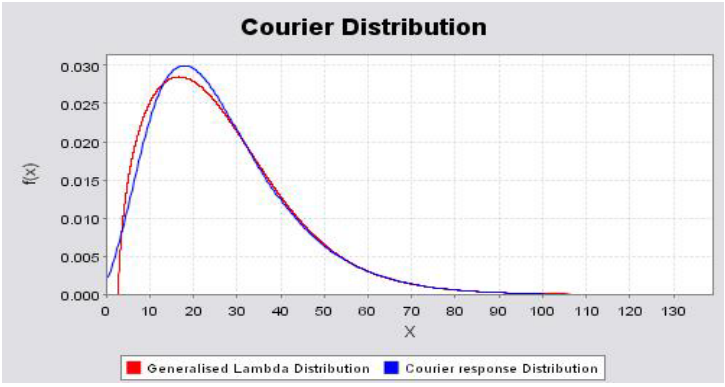


Fig. 8.7

The GLD provides a decent approximation, with the GLD being slightly lower at the peak, and the initial x value (representing time) being a bit larger than that of the actual distribution. The cdfs we get are shown in Fig 8.8.

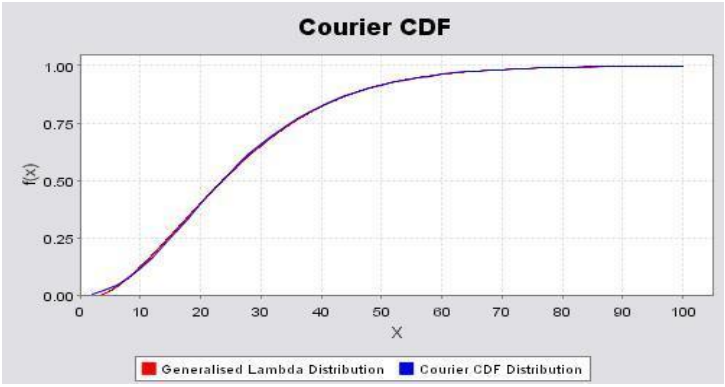


Fig.8.8

The GLD cdf seems to provide an even better approximation than it did for the pdf we can see that again as expected the initial x value of the GLD is larger than the actual initial value, but other than that it is hard to distinguish between the two cdfs.

The exact pdf and cdf was calculated in 134 seconds, while it took 0.66 seconds to generate the first four moments together with the GLD parameters taking an average of 280.6 milliseconds to calculate, meaning that the approximation took a total of 0.9406 seconds to generate.

The second GSPN model we look at is called FMS see Ciardo and Trivedi (1993). This is a 22-place model of a flexible manufacturing system composed of three types of machines and four types of parts. The time returned is that for there being 4 unassembled parts at the start of the passage to the completion of the first finished product.

The FMS model has the following first four raw moments:

(6.81434, 56.5869, 560.595, 6519.26)

This yields the GLD approximation as shown in Fig 8.9.

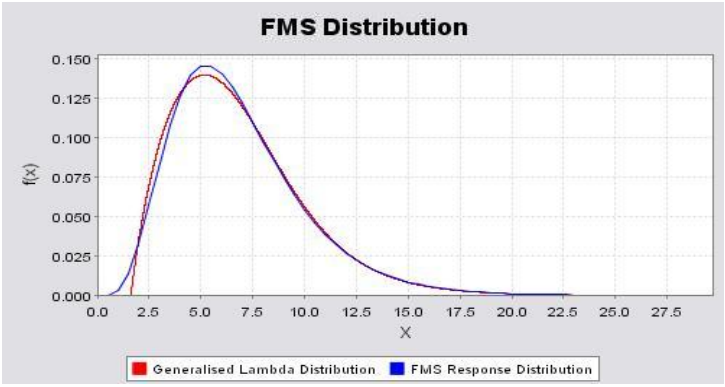


Fig 8.9

Again we see that the GLD approximation is slightly lower at the peak and has a larger initial x value, but despite this you can clearly see that the GLD provides a very good approximation.

The cdf we get is plotted in Fig. 8.10 against the actual FMS cdf

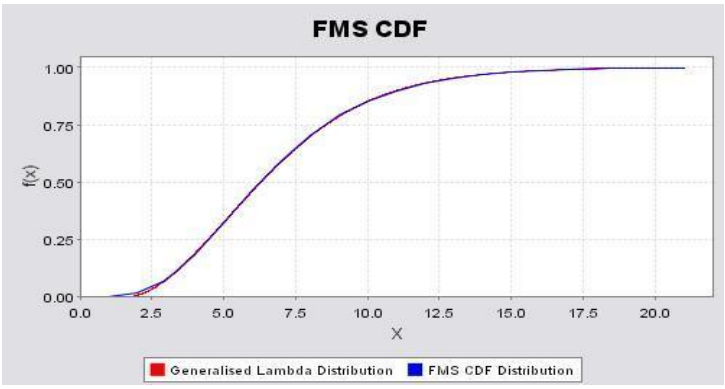


Fig 8.10

This is a 35,910 tangible state model, which using the Laplace-transform based technique took 834 seconds (13 minutes and 54 seconds) to calculate the pdf and cdf. The corresponding first four moments took 7.64 seconds while the GLD parameters takes an average of 0.4188, meaning that the total time taken to obtain the approximation takes almost 100 times less at 8.0588 seconds.

8.3 Queuing Network Model

Up to now we have seen that the GLD has provided very good approximations to all the distributions and response time densities I have compared it to. We shall in this section compare the GLD to a queuing network model for which the GLD approximation is not as satisfactory. We shall also see here that despite restricting the accuracy range as discussed in section 6.4 we still get a few sets of GLD parameters returned.

The model we shall investigate is called the tree model which models the closed tree-like network shown in Fig. 8.11 with 8 customers. The cycle time is that for a tagged customer arriving at the back of the first queue to returning to the queue, see Harrison and Knottenbelt (2002).

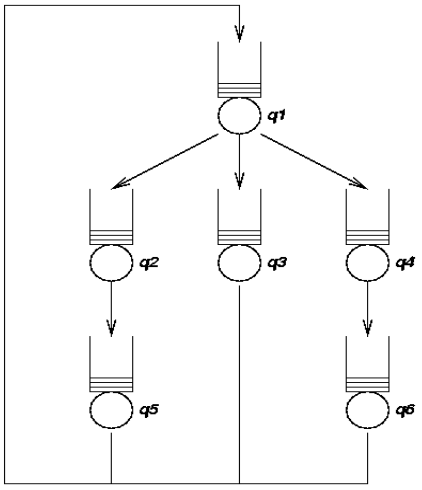


Fig. 8.11

The first four raw moments of the tree model are:

(2.82125, 12.6453, 80.2427, 652.093).

Putting these values into my program returns four sets of GLD parameters each, in the accuracy (denoted by $\sqrt{\varepsilon}$ see section 6.4) band of:

$$1.0 \times 10^{-4} < \sqrt{\varepsilon} < 1.0 \times 10^{-10}$$

The four sets of parameters we get are:

$\lambda_1 = 1.2256695, \lambda_2 = 0.4741358, \lambda_3 = 3.5654279, \lambda_4 = 0.02505288.$ (a)

$\lambda_1 = 0.0846688, \lambda_2 = 0.4346116, \lambda_3 = 8.9749451, \lambda_4 = 0.04338539.$ (b)

$\lambda_1 = 6.9899145, \lambda_2 = 0.0108904, \lambda_3 = 9.5705970, \lambda_4 = 19.3236601.$ (c)

$\lambda_1 = 2.5813382, \lambda_2 = 1.0779538, \lambda_3 = -0.0010101, \lambda_4 = -0.2061157.$ (d)

Since the most often returned set of parameters are set (a), and it is this set that seems to give the best fit, I shall be investigating how well the GLD using the set (a) parameters fit the tree response time density.

I have included in Appendix B Figures B.1 to B.6, the graphs of how the pdfs and cdfs of the GLDs when using the parameters shown by (b), (c) and (d) compare to the actual distribtion. If you have a look at these graphs we see that the GLDs returned by set (a) and (b) are very similar with the GLD from (b) having a slightly lower initial f(x) value. However the GLDs returned by set (c) and (d) are markedly different.
The fit provided by the GLD using set (a) is shown in Fig. 8.12.

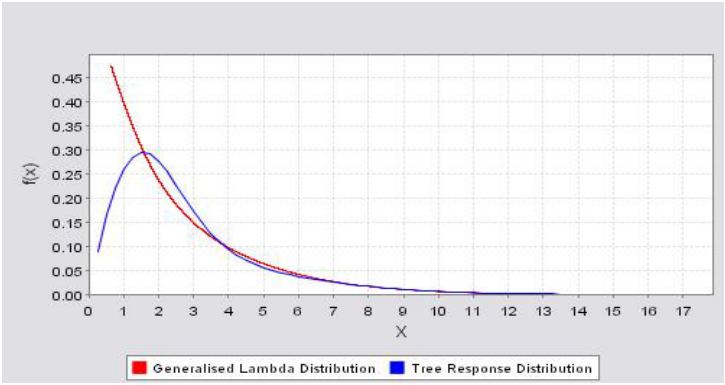


Fig. 8.12

We can see that curve returned by the GLD is very different from that of the actual distribution, the GLD returns a J-shaped curve whilst the actual distribution is a unimodal distribution with a truncated left tail. However we can clearly see that although initially the GLD doesn't provide a good fit, once the GLD crosses the actual distribution at around x = 1.6 the GLD does give you a favourable approximation.

I am unsure as to why the GLD fit isn't as impressive as the previous examples, the GLD fit belongs to the Class III family of GLD shapes, however the actual distribution is more similar to the Class Ib family of shapes.

However despite the GLD approximation to the pdf not being as good as before, the GLD cdf approximation, Fig. 8.12, does return a favourable fit which is useful if you wish to use the GLD to approximation to the cdf to find say for instance the 95th percentile. On average it took 0.6106 seconds to compute the GLD parameters.

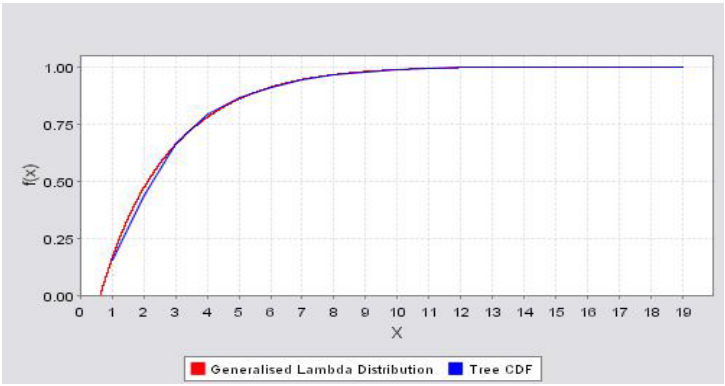


Fig. 8.14

In order to check whether or not this GLD pdf approximation failure is particular to just this example of the tree model or to the whole family of tree models, I have applied the GLD to a few other examples of the tree model, some with 6 customers and others with different rates. However I have found that I still get the J-shaped approximation to the actual unimodal approximation, but again the cdf approximations are good.

8.4 Bimodal Distributions

Although the GLD family of curves doesn't included bimodal curves, I think that it would be interesting to see how well the GLD does perform in approximating bimodal distributions.

The bimodal distribution I shall be investigating is called Branch see Fig 8.14. This is a branching Erlang model with two equiprobable branches, one of which is an Erlang (3, 1) delay and the other of which results in a Erlang (12, 1) delay (details in Harrison and Knottenbelt (2002). The response time is the time taken for the token to get from the place it is on in Fig. 8.14, back to that place.

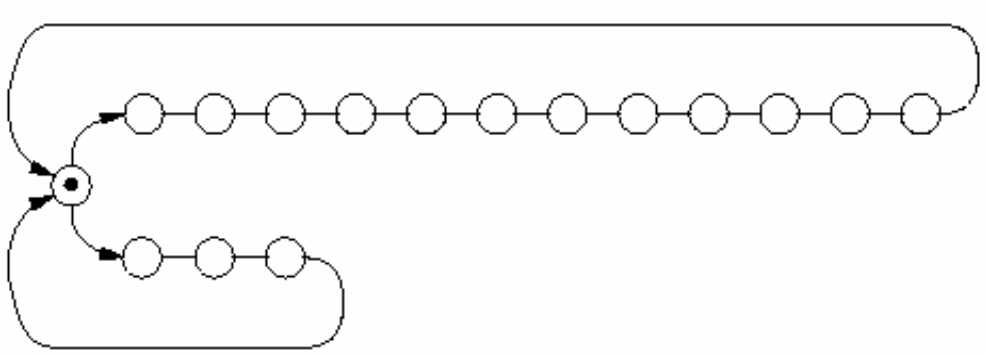


Fig. 8.14

The first four raw moments for Branch are: (8.5, 101, 1425, 22260).

The GLD approximation we get is shown in Fig. 8.15.

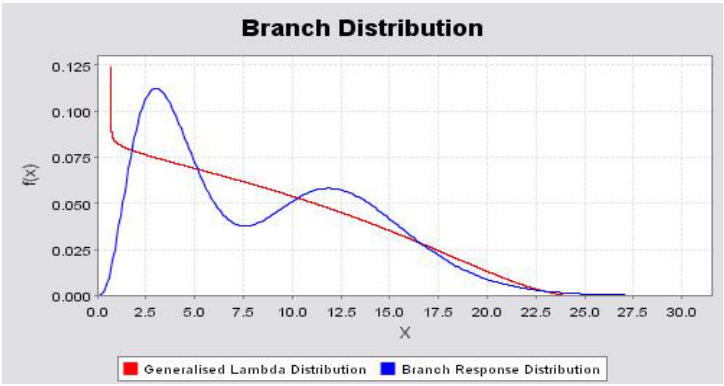


Fig. 8.15

As expected we see that the GLD doesn't really follow the actual distribution very closely, however as a unimodal approximation it doesn't seem to be a bad attempt. This is confirmed when we look at the cdf see Fig. 8.16, for which we get a surprisingly good fit, with the GLD parameters taking an average of 0.2684 seconds to compute.

We see that the actual cdf wavers slightly above and below the GLD cdf approximation, which is what you would expect. However you can clearly see that the GLD provides a pretty good approximation, especially if you wish to find say the 95th percentile, the GLD would return a fairly accurate result.

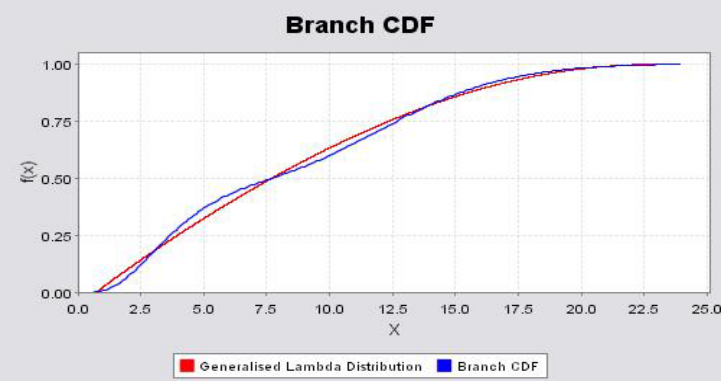


Fig. 8.16

Although this GLD approximation to the branch distribution provides the seemingly worst fit to all of the cdfs I have investigated, when you compare this fit to the bounds returned by the WinMoment tool Fig 8.17, you see that the GLD approximation is a substantial improvement.

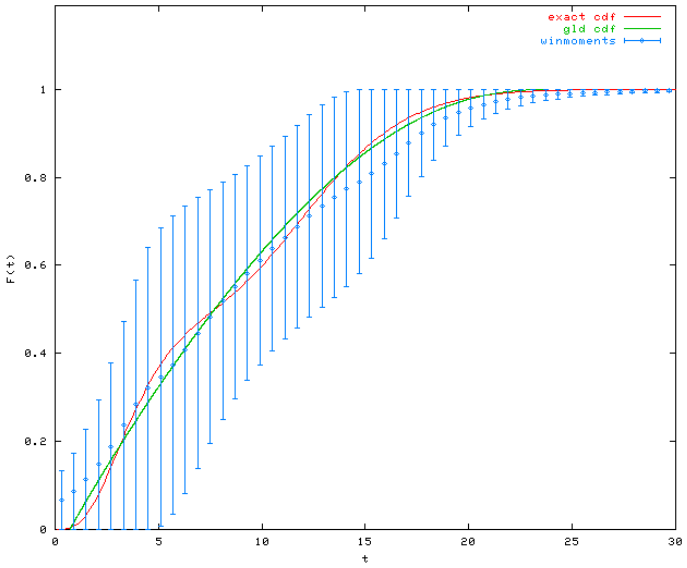


Fig 8.17

If we know look at another bimodal distribution, called the Branch 2 model which is as described as above this time with the upper branch having Erlang (3, 1) delay, and the bottom having Erlang (12, 2) delay, which brings the two peaks of the response time density closer together to form an almost unimodal curve.

The raw moments of the Branch 2 model are: (5.5, 36.5, 276, 2307.75).

The GLD returns a unimodal approximation to the bimodal density see Fig. 8.18, where the parameters take on average 0.1244 seconds to be found.

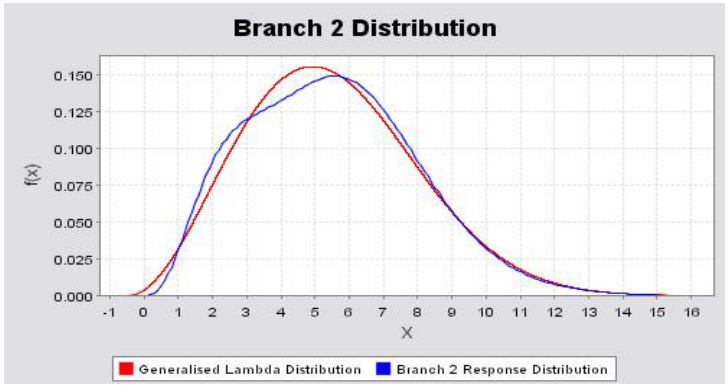


Fig. 8.18

We see that the unimodal approximation provides a fairly good fit, with the GLD having a higher peak than either of the two peaks of the actual distribution and the GLD having a slightly negative initial x value, which isn't ideal when x represents time!

When we look at the cdf the GLD again gives a very good approximation see Fig. 8.19.

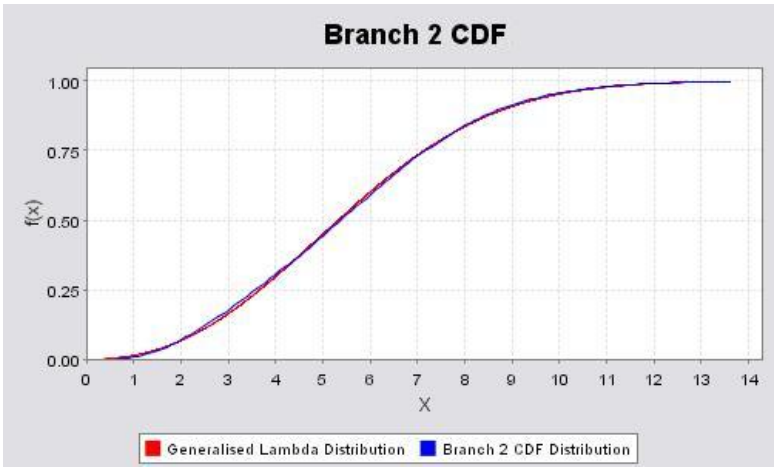


Fig. 8.19

From these two examples we can see that even though the GLD family doesn't contain a family of bimodal curves, the GLD does provide a fairly good unimodal approximation to bimodal distributions, especially in regard to the cdf and especially if the peaks in the bimodal distribution are fairly close together.

9. Sensitivity to the Accuracy of Moments

In order to investigate the effect that inaccuracy in raw moments, has on the approximation derived from the GLD, I shall look at the effect it has on the approximations to the Standard Normal distribution.

From Section 7.1 we know that the mean, variance , skewness and kurtosis of N(0,1) are:

$$\mu = 0, \quad \sigma^2 = 1, \quad \alpha^3 = 0, \quad \alpha^4 = 3.$$

The first four raw moments happen to be exactly the same (0, 1, 0, 3).
The GLD pdf approximation is accurate to +/- 0.0028130386 whilst the cdf is accurate to +/- 0.0011804758.

I shall now be increasing each of the raw moments by 0.001, 0.01 and 0.1, to see what the maximum absolute difference between this and N(0,1).

New moments	Max diff. of pdf	Diff. to GLD(0,1,0,3)	Max diff. of cdf	Diff. to GLD(0,1,0,3)
(0.001, 1.001, 0.001, 3.001)	0.0023399	-0.0004731	0.000989	-0.00019141
(0.01, 1.01, 0.01, 3.01)	0.0051945	0.0023815	0.007927	0.006747143
(0.1, 1.1, 0.1,3.1)	0.144317	0.0562246	0.1031995	0.10201906

As you can see the moments can be +0.01 across the board and yet only cause a maximum difference of 0.0051945 to the actual N(0,1) pdf while the cdf approximation is only affected by a maximum of 0.006747143.

This seems to indicate that the GLD is fairly stable and that if you can generate the raw moments to an accuracy of +/- 0.001, you can be confident of obtaining a distribution that has a maximum difference of +/- 0.0005 to the pdf you would obtain if you were to have exact moments, indeed it seems that in this case with an error of +0.001 in all the moments you get a better approximation to N(0,1) than if you used exact moments, if you look at the graphs for GLD using (0.001, 1.001, 0.001, 3.001) against GLD using (0, 1, 0, 3) you cannot distinguish between the two.

Graphs illustrating the difference in approximation to the original GLD using (0, 1, 0, 3) that the values of the moments shown in the above table return in both the pdf and cdf are shown in Appendix C.

10. Conclusions and Future Work

I have implemented and described in detail the use of the Generalized Lambda Distribution to match distributions and data via the method of moments. I have also investigated how well the GLD approximations compare to the exact distributions as well as against the current available tool.

As demonstrated in Chapters 7 and 8, the GLD provides a good approximation to a range of standard probability distributions and also to a wide range of response time models, particularly in approximating unimodal distributions. Even when the GLD pdf approximation isn't very good see section 8.3, the corresponding cdf seems to be satisfactory. The GLD even seem to be effective in approximating bimodal distributions for which results are favourable.

We have seen that this method of approximating response time densities returns some very good results in particular the cdfs returned are a particularly good fit. This all means that by using the GLD to approximate response time densities we use substantially less computing power and time with little loss in accuracy.

In fact using the GLD and the first four moments to determine approximate response time densities in Markov and Semi-Markov stochastic models, requires two orders of magnitude less computation than exact conventional Laplace transform-based techniques.

We have however seen that there is a slight drawback in using the GLD to approximate distributions, this is the fact that there do sometimes occur more than one set of GLD parameters which provide differing approximations. There is no way of knowing through using my program whether or not there are other solutions, and the current best way of ensuring that the best approximation will be obtained is to run the program a few times for the given moments to see which is the best approximation.

Despite this drawback, though implementing the Generalized Lambda Distribution, my tool provides an efficient and seemingly relatively stable method in producing an accurate approximation to response time densities which can be readily and easily used, that returns significantly better results than currently available tools. Indeed the closeness of these approximations has led to the findings of this project being presented in the paper shown in Appendix D (Efficient Approximation of Response Time Densities and Quantiles in Stochastic Models).

There is much scope for further investigation, first a fuller investigation as to how well the GLD approximates response times densities should be undertaken, by testing the GLD against more models, especially in relation to bimodal models.

Another area which should be looked at is why there are some models which are not approximated very well at all; in particular the reasons why the GLD fails to approximate the Tree-like network model as discussed in Section 8.3.

This may lead to a study as to why there is sometimes more than one possible set of values to the GLD parameters, for what values of skew and kurtosis for which this does occur and which set would be the most appropriate to use. The results obtained could then be applied to my program to ensure the best approximation is always returned

It would also be helpful if there was some work in providing a method of quantifying the accuracy of the GLD approximations when it is applied to data or densities for which there are no fully defined probability density functions.

Other extensions to this project may include adding on other distributions in order to approximate bimodal distributions and other such distributions for which the GLD fails to produce an adequate approximation.

Improvements to my tool may include using a more accurate method to approximate the cdf. as the current trapeze method, though simple to implement, is not the most accurate method you can use. You could also look into extending the graphical user interface so that it also displays the parameters to the GLD and the exact accuracy to which the moments are matched.

Lastly since my sensitivity study is restricted to only looking at the Standard Normal Distribution, there should also be a more thorough and in depth sensitivity study.

11. User Guide

There are four simple steps to using my tool which approximates the distribution given its first four moments:

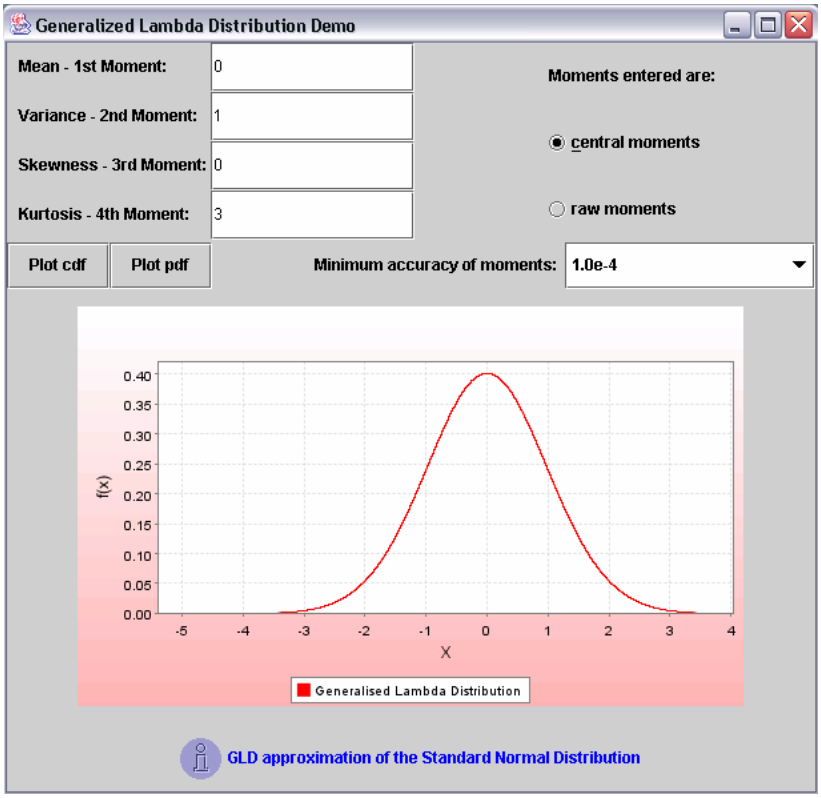
1. Enter the first four moments.
2. Indicate the type of moments entered.
3. Set the desired level of accuracy.
4. Click on either the pdf. or the cdf plot button.

My tool allows the user to input the mean, variance, skewness and kurtosis (central moments) or the first four raw moments of the data/distribution you wish to approximate.

You may choose the desired minimum level of accuracy to which the moments are to be matched to i.e. the value of $\sqrt{\varepsilon}$ as discussed in section 6.2. This functionality is there to allow you to search for the GLD parameters when a previous search to a higher level of accuracy has failed, however in general it is recommended that you set the accuracy to at least the default level of 1.0×10^{-4} since running the program at a lower level of accuracy may return incorrect parameter values which in turn leads to an inaccurate approximation.

The tool will return error messages if the first four moments are not inputted as either integers or decimals, it does not read in fractions.

It will also return an error message if it cannot find the parameters to the set level of accuracy.



Screen capture of my tool to approximate densities from the first four moments.

Bibliography

Abramovitz, M. and I. A. Stegun (1964), Handbook of Mathematical Functions, Dover.

Akhiezer, N. I. (1965), The Classical Moment Problem and some related questions in analysis, Oliver & Boyd, London.

Besset, D. H (2001), Object-Orientated Implementation of Numerical Methods - An Introduction with Java and Smalltalk, Morgan Kaufmann.

Bradley, J. T., N. J. Dingle, W.J. Knottenbelt and H.J. Wilson (2003), Hypergraph-based parallel computation of passage time densities in large semi-markov models, In Proc 4th International Meeting on the Numerical Solution of Markov Chains (NSMC '03), pp 99-120. Urbana-Champaign, Illinois.

Ciardo, G. And K.S. Trivedi. A decomposition approach for stochastic reward net models. Performance Evaluation, 18(1): 37-59.

Dudewicz, E. J. and Z. A. Karian (1996), The Extended Generalized Lambda Distribution (EGLD) System For Fitting Distributions To Data With Moments,

Freimer, M., G. Mudholkar, G.Kollia and C. Lin (1988), A study of the generalized Tukey Lambda family, Communications in Statistics, Theory and Methods, 17(10): 3547-3567.

Gautama, H. and A. J. C. van Gemund (2002), Towards Performance Estimation of Data-Dependent Task Parallel Compositions, Proc 18th UK Performance Engineering Workshop (UKPEW 2002) Glasgow UK p81-92.

Harrison, P.G. and W. J. Knottenbelt (2002), Passage time distributions in large Markov chains, In Proc. ACM SIGMETRICS 2002, Marina Del Ray, California.

Hastings, C., F. Mosteller, J.W. Tukey, and C.P. Winsor (1947), Low Moments for Small Samples: A Comparative Study of Statistics, Annals of Mathematical Statistics, 18, 413-26.

Jaynes, E. T. (1978), Where do we stand on maximum entropy, in R. D. Levine, M. Tribus (editors). The Maximum Entropy Formalism, MIT Press, Cambridge, MA 15-118.

Karian, Z. and E. Dudewicz (2000), Fitting Statistical Distributions: The Generalized Lambda Distribution and Generalized /Bootstrap Methods, Boca Raton: CRC Press.

Koza, J. R., Forrest H, Bennett III, David Andre, and Martin A. Keane (1999), Genetic Programming III, Morgan Kaufmann.

Lakhany, A. and H. Mausser (2000), Estimating the Parameters of the Generalized Lambda Distribution, Algo Research Quarterly, 3(3): 47- 58.

Nelder, J.A. and R. Mead (1965), A simplex method for function minimization, Computer Journal, 7: 308-313.

Powell, M. J. D. (1962), An iterative method for finding stationary values of a function of several variables, Computer Journal, 5(2): 147-151.

Press, W. H., Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery (1992), Numerical Recipes in C: The Art of Scientific Computing, Cambridge University Press.

Rácz, S. (2002), Numerical analysis of communication systems through Markov reward models, PhD thesis, chapter 5.Budapest University of Technology and Economics.

Ramberg, J. S., E. Dudewicz, P. Tadikamalla and E. Mykytka (1979), A probability distribution and its uses in fitting data, Technometrics, 21(2): 201-214.

Ramberg, J. S. and B. W. Schmeiser, (1974), An approximate method for generating asymmetric random variables, Communications of the ACM, 17(2): 78-82.

Woodside, C.M. and Y. Li (1991), Performance Petri net analysis of communication protocol software by delay-equivalent aggregation, In Proc of the 4th International Workshop on Petri nets and Performance Models, p64-3. Melbourne Austrailia.

Appendix A – Details of Numerical Methods

Lanczos approximation of the Gamma function

The Lanczos approximation, taken from Press et al (1992) is as follows

$$\Gamma(x) \approx e^{x+5/2} (x + 5/2) \frac{\sqrt{2\pi}}{x} \left[c_0 + \sum_{n=1}^{6} \frac{c_n}{x + n} \right]$$

where

c_1 =1.000000000190015
c_2 = 76.18009172947146
c_3 = -86.50532032947146
c_4 = -1.231739572450155
c_5 = $1.208650973866179 \times 10^{-3}$
c_6 = $-5.395239384953 \times 10^{-6}$

Approximation to the Standard Normal cdf

This approximation taken from Abramovitz and Stegun (1964), which is accurate to a least 7.5×10^{-8}

$$F(x) \approx \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \sum_{i=1}^{5} a_i r(x)^i \quad \text{for } x \geq 0$$

where $r(x) = \frac{1}{1 - 0.2316419 x}$

and

a_1 = 0.31938153
a_2 = -0.356563782
a_3 = 1.7814779372
a_4 = -1.821255978
a_5 = 1.330274429

to find the negative values we use the fact: $F(x) = 1 - F(-x)$

Incomplete Gamma Function

I have implemented the method shown in Besset (2000) to find the value of the Incomplete Gamma Function defined as

$$\Gamma(x, \alpha) = \frac{1}{\Gamma(\alpha)} \int_0^x t^{\alpha-1} e^{-t} dt$$

where $\Gamma(\alpha)$ is the Gamma Function

The previous integral can be expressed as the following infinite (series see Abramovitz & Stegun (1964)).

$$\Gamma(x, \alpha) = \frac{e^{-x} x^{\alpha}}{\Gamma(\alpha)} \sum_{n=0}^{\infty} \frac{\Gamma(\alpha) x^n}{\Gamma(\alpha + 1 + n)}$$

This series converges well for $x < \alpha + 1$.

The incomplete gamma function can also be written as (see Abramovitz & Stegun (1964)).

$$\Gamma(x, \alpha) = \frac{e^{-x} x^{\alpha}}{\Gamma(\alpha)} \frac{1}{F(x - \alpha + 1, \alpha)}$$

where $F(x , \alpha)$ is the continued fraction

$$F(x, \alpha) = \left(x + \frac{1(\alpha - 1)}{x + 2 +} \frac{2(\alpha - 2)}{x + 4 +} \frac{3(\alpha - 3)}{x + 6 +} \ldots \right)$$

This continued fraction converges well for $x > \alpha + 1$ which is where the infinite series didn't converge well. Therefore by employing one of the two above methods we can calculate the incomplete gamma function. For a full implementation see Besset (2000) or Press et al (1992).

Appendix B – Tree Distribution Graphs

As discussed in Section 8.3 the tree-like distribution moments return a number of different sets of GLD parameters. Plots of how when used to return a GLD, the other sets of GLD parameters (b), (c), and (d) compare to the actual distribution are shown here.

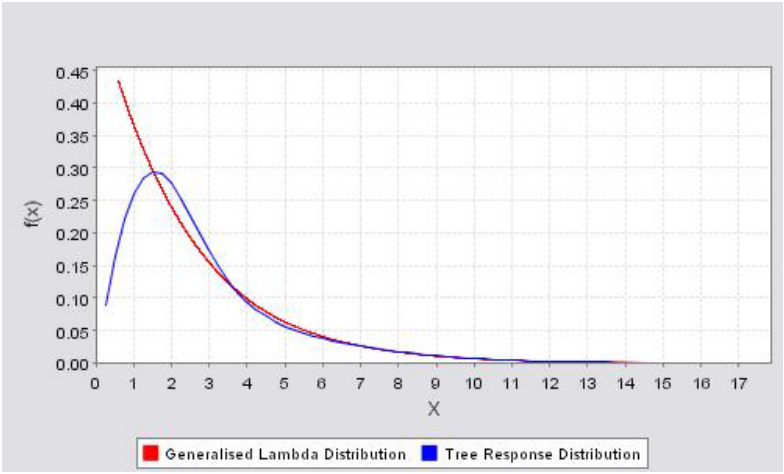


Fig B.1 pdf of GLD with parameters set (b) and actual tree density.

Fig B.2 cdf of GLD with parameters set (b) and actual tree cdf

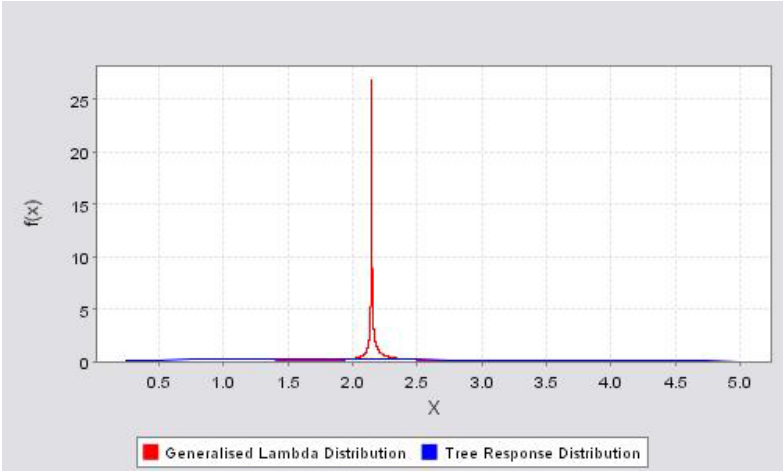


Fig B.3 pdf of GLD with parameters set (c) and actual tree density.

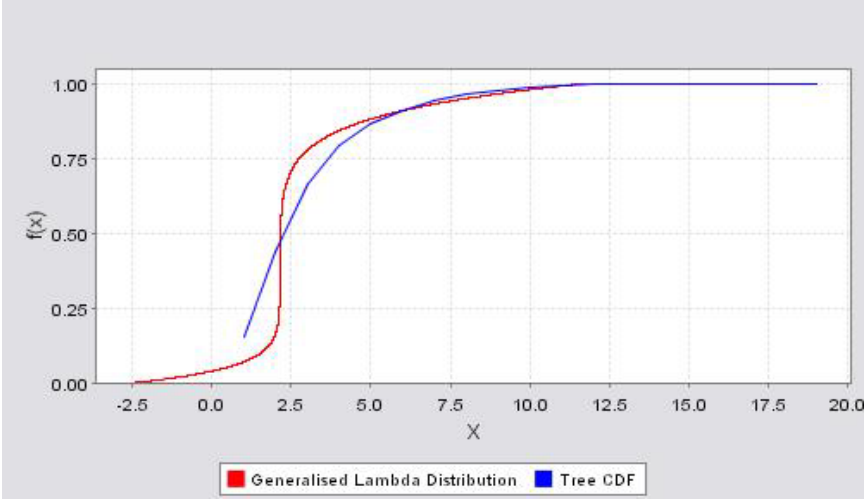


Fig B.4 cdf of GLD with parameters set (c) and actual tree cdf

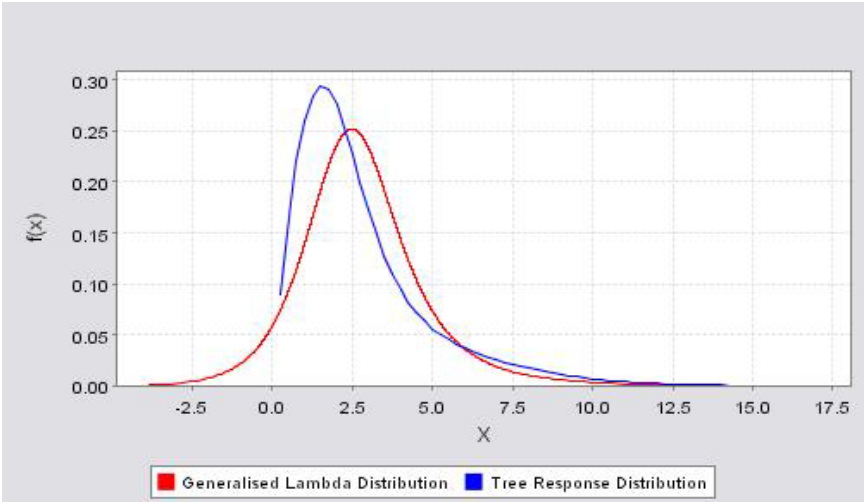


Fig B.5 pdf of GLD with parameters set (d) and actual tree density.

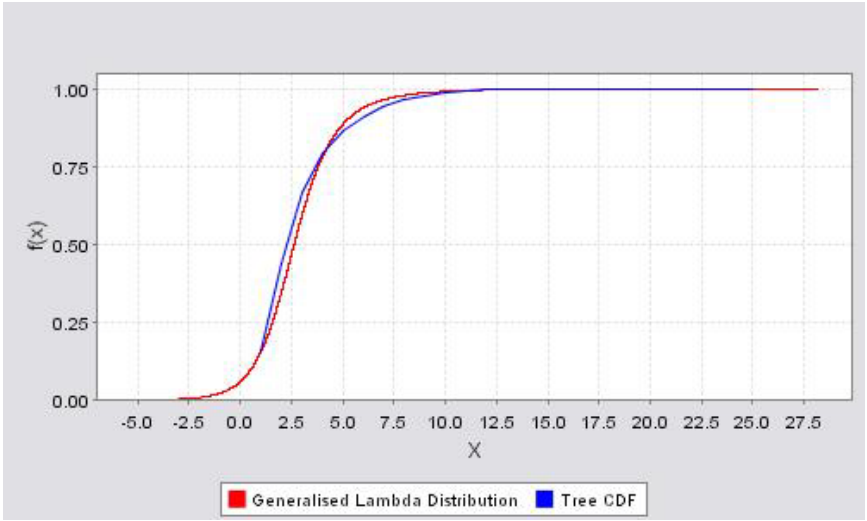


Fig B.6 cdf of GLD with parameters set (d) and actual tree cdf

Appendix C – Graphs from Sensitivity study

The following six graphs are from the sensitivity study, plotting both the pdf and cdf for GLD using raw moments (0, 1, 0, 3) against the GLDs using (0.001, 1.001, 0.001, 3.001), (0.01, 1.01, 0.01, 3.01) and (0.1, 1.1, 0.1, 3.1).

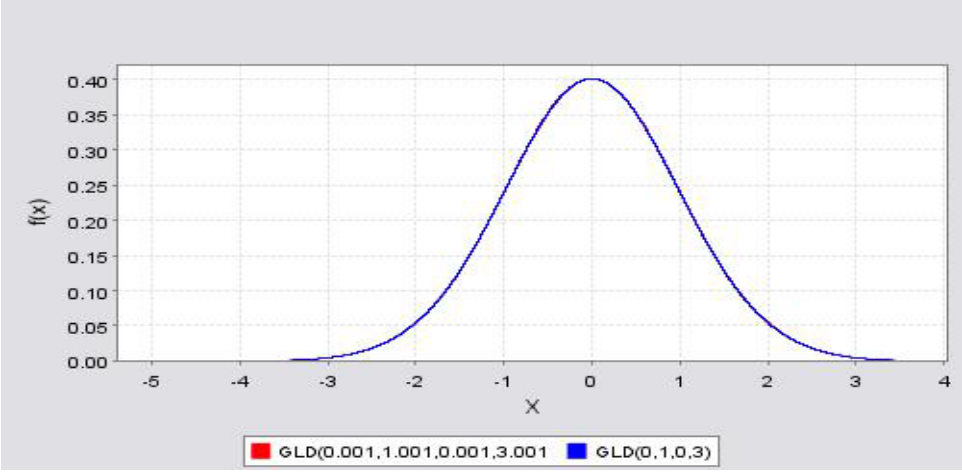


Fig. C.1 pdf of GLDs using (0, 1, 0, 3) against (0.001, 1.001, 0.001, 3.001)

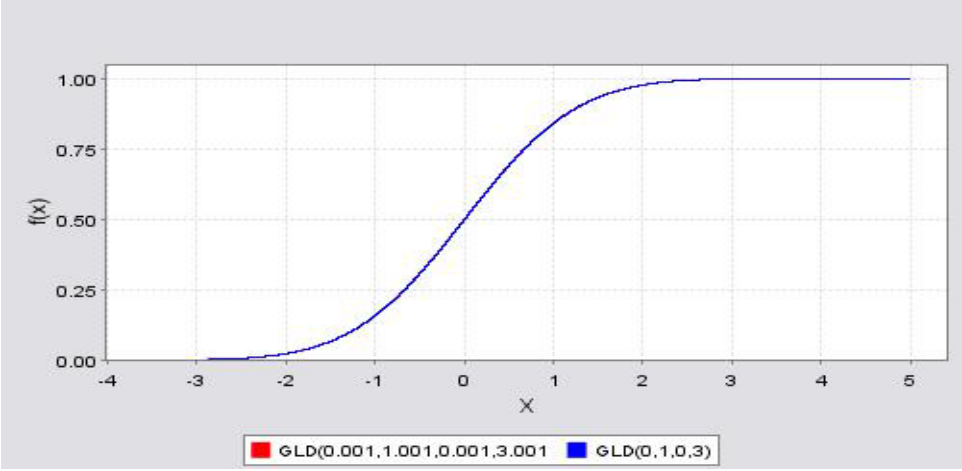


Fig. C.2 cdf. of GLDs using (0, 1, 0, 3) against(0.001, 1.001, 0.001, 3.001)

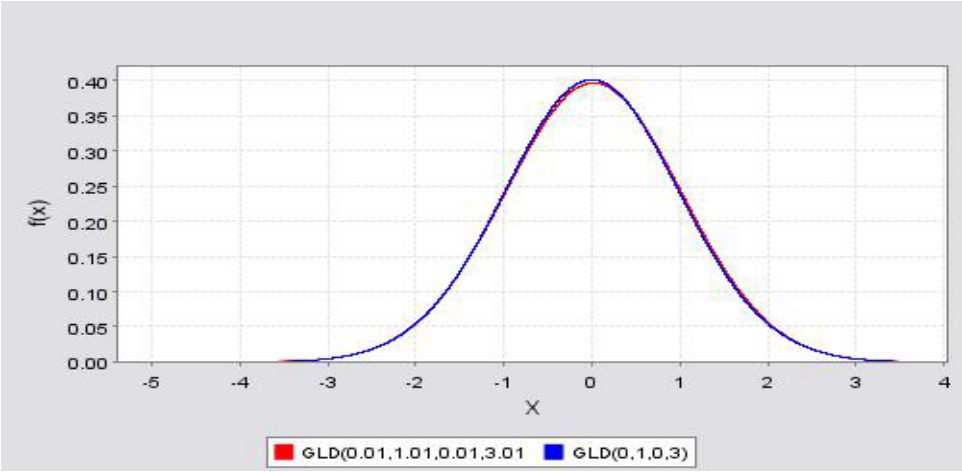


Fig. C.3 pdf of GLDs using (0,1) against (0.01, 1.01, 0.01, 3.01)

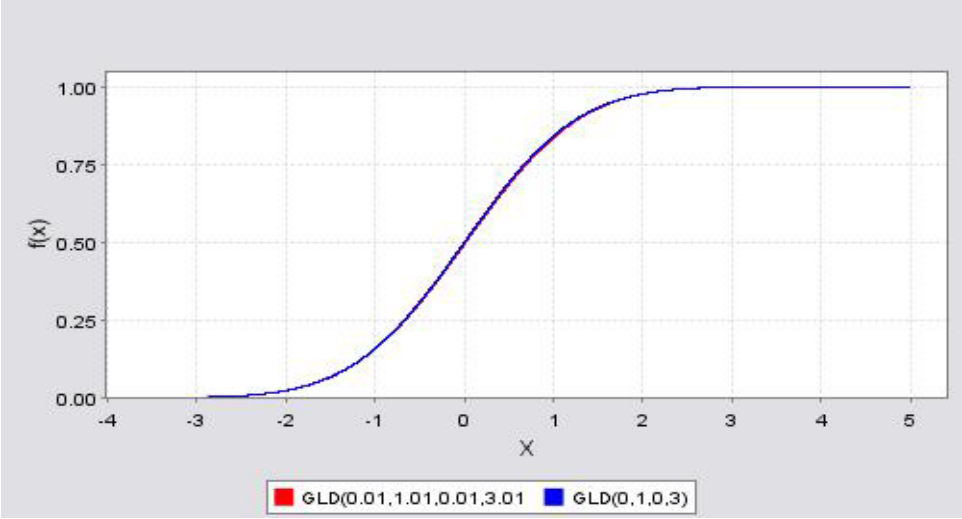


Fig. C. 4 cdf of GLDs using (0,1) against (0.01, 1.01, 0.01, 3.01)

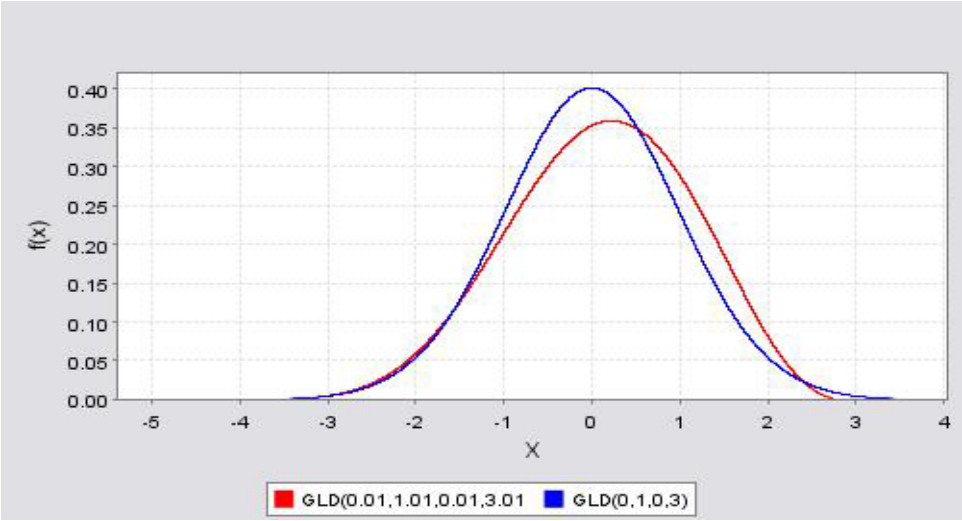


Fig. C.5 pdf of GLDs using (0,1) against (0.1, 1.1, 0.1, 3.1)

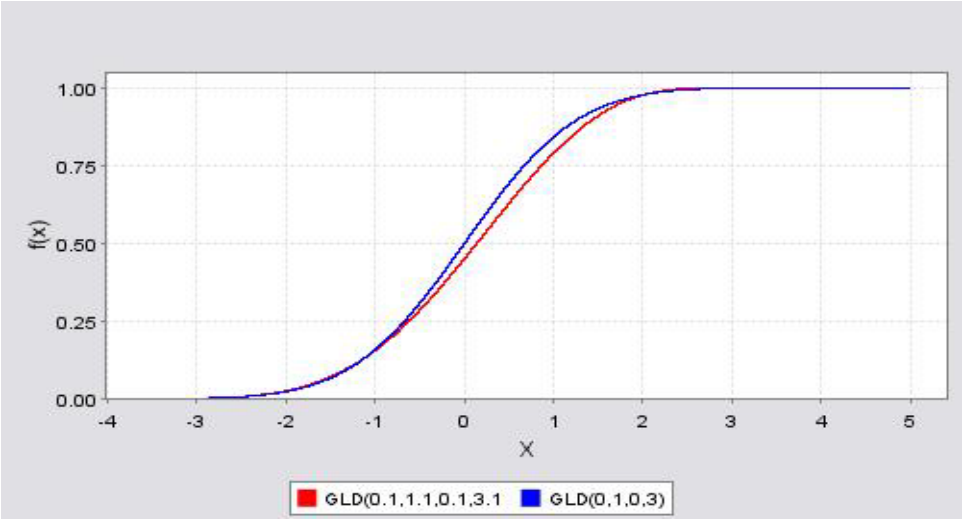


Fig. C.6 pdf of GLDs using (0,1) against (0.1, 1.1, 0.1, 3.1)

Appendix D

(Efficient Approximation of Response Time Densities and Quantiles in Stochastic Models)

Efficient Approximation of Response Time Densities and Quantiles in Stochastic Models

Susanna W.M. Au-Yeung, Nicholas J. Dingle and William J. Knottenbelt

August 31, 2003

Abstract

Response time densities and quantiles are important performance and quality of service metrics, but their analytical derivation is, in general, very expensive. This paper presents a technique for determining approximate response time densities in Markov and semi-Markov stochastic models that requires two orders of magnitude less computation than exact Laplace transform-based techniques. The method computes the first four moments of the desired response time and then makes use of Generalised Lambda Distributions to obtain an approximation of the corresponding density. Numerical results show good agreement over a range of response time curves, particularly for those that are unimodal.

1 Introduction

Software and hardware system architects are increasingly required to consider response time guarantees as key quality of service metrics. Indeed, response time quantiles are routinely specified in Service Level Agreements (SLAs) and it is therefore important to quantify the risk of violating response time targets. Stochastic models can help to address this problem, even before the implementation phase. In particular, analytical methods based on the numerical inversion of Laplace transforms have recently been developed to extract response time densities and quantiles from high-level stochastic modelling formalisms based on Markov and semi-Markov chains [8, 5, 3]. However, these methods are computationally expensive, and large models require the availability of a cluster of workstations to calculate results in reasonable time. For example, it requires over 15 hours processing time on a 64-node cluster to solve for a response time density in a 10.9 million state semi-Markov chain model of a voting system [3].

This paper addresses this problem by presenting a low-cost technique which seeks to approximate response time densities and quantiles from the corresponding first four moments. In contrast to the exact technique which requires the solution of many hundreds of sets of linear equations, calculation of the moments requires the solution of just four sets. In both cases the dimension of the linear equations is given by the number of states in the model. A Generalised Lambda Distribution (GLD) is then fitted to these moments as an approximation to the exact density. Fitting the moments is a rapid operation and its complexity does not depend on the number of states in the model. The corresponding cumulative distribution function (which can be used to determine response time quantiles) is then obtained by numerical integration of the GLD.

The rest of this paper is organised as follows. Section 2 outlines methods for the computation of response time densities and their corresponding moments in Markov and semi-Markov models. Section 3 introduces Generalised Lambda Distributions and describes the process of fitting a GLD to a given set of moments. Section 4 compares exact and approximate response time densities for a range of models, including Generalised Stochastic Petri nets (GSPNs), Markovian queueing networks and Semi-Markov Stochastic Petri nets (SMSPNs). Both uni- and bimodal densities are considered. Finally, Section 5 concludes and discusses opportunities for future work.

2 Response Time Analysis

2.1 Semi-Markov Processes

Consider a Markov renewal process $\{(X_n, T_n) : n \geq 0\}$ where T_n is the time of the nth transition ($T_0 = 0$) and $X_n \in \mathcal{S}$ is the state at the nth transition. Let the kernel of this process be:

$$R(n, i, j, t) = \mathbb{P}(X_{n+1} = j, T_{n+1} - T_n \leq t \mid X_n = i)$$

for $i, j \in \mathcal{S}$. The continuous time semi-Markov process (SMP), $\{Z(t), t \geq 0\}$, defined by the kernel R, is related to the Markov renewal process by:

$$Z(t) = X_{N(t)}$$

where $N(t) = \max\{n : T_n \leq t\}$, i.e. the number of state transitions that have taken place by time t. Thus $Z(t)$ represents the state of the system at time t. We consider time-homogeneous SMPs, in which $R(n, i, j, t)$ is independent of any previous state except the last. Thus R becomes independent of n:

$$\begin{aligned} R(i, j, t) &= \mathbb{P}(X_{n+1} = j, T_{n+1} - T_n \leq t \mid X_n = i) \\ &= p_{ij} H_{ij}(t) \end{aligned}$$

where $p_{ij} = \mathbb{P}(X_{n+1} = j \mid X_n = i)$ is the state transition probability between states i and j and $H_{ij}(t) = \mathbb{P}(T_{n+1} - T_n \leq t \mid X_{n+1} = j, X_n = i)$, is the sojourn time distribution in state i when the next state is j.

2.2 First passage times

Consider a finite, irreducible, continuous-time semi-Markov process with N states $\{1, 2, \ldots, N\}$. Recalling that $Z(t)$ denotes the state of the SMP at time t ($t \geq 0$), the first passage time from a source state i at time t into a non-empty set of target states $\vec{j}$ is:

$$P_{i\vec{j}}(t) = \inf\{u > 0 : Z(t + u) \in \vec{j}, N(t + u) > N(t), Z(t) = i\}$$

For a stationary time-homogeneous SMP, $P_{i\vec{j}}(t)$ is independent of t and we have:

$$P_{i\vec{j}} = \inf\{u > 0 : Z(u) \in \vec{j}, N(u) > 0, Z(0) = i\}$$

$P_{i\vec{j}}$ has an associated probability density function $f_{i\vec{j}}(t)$. In general, the Laplace transform of $f_{i\vec{j}}$, $L_{i\vec{j}}(s)$, can be computed by solving a set of N linear equations:

$$L_{i\vec{j}}(s) = \sum_{k \notin \vec{j}} r^*_{ik}(s) L_{k\vec{j}}(s) + \sum_{k \in \vec{j}} r^*_{ik}(s) \qquad : \text{for } 1 \leq i \leq N \tag{1}$$

where $r^*_{ik}(s)$ is the Laplace-Stieltjes transform (LST) of $R(i,k,t)$ from Section 2.1 and is defined by:

$$r^*_{ik}(s) = \int_0^\infty e^{-st}\, \mathrm{d}R(i,k,t)$$

When there are multiple source states, denoted by the vector $\vec{i}$, the Laplace transform of the passage time density at steady-state is:

$$L_{\vec{i}\vec{j}}(s) = \sum_{k \in \vec{i}} \alpha_k L_{k\vec{j}}(s)$$

where the weight α_k is the probability at equilibrium that the system is in state $k \in \vec{i}$ at the starting instant of the passage. If $\tilde{\pi}$ denotes the steady-state vector of the embedded discrete-time Markov chain (DTMC) with one-step transition probability matrix $P = [p_{ij}, 1 \leq i,j \leq N]$, then α_k is given by:

$$\alpha_k = \begin{cases} \pi_k / \sum_{j \in \vec{i}} \pi_j & \text{if } k \in \vec{i} \\ 0 & \text{otherwise} \end{cases}$$

2.3 Moments

Let $M_{i\vec{j}}(n)$ denote the nth moment of the first passage time between a given source state i and set of target states $\vec{j}$, and let $m_{ik}(n)$ denote the nth moment of the holding time in state i with next state k. Assuming the derivatives of $r^*_{ik}(s)$ exist at the origin, we have

$$m_{ik}(n) = (-1)^n \left. \frac{\mathrm{d}^n r^*_{ik}(s)}{\mathrm{d}s^n} \right|_{s=0}$$

Hence, using Leibnitz' rule,

$$\begin{aligned} M_{i\vec{j}}(n) &= \sum_{k \notin \vec{j}} \sum_{r=0}^{n} \binom{n}{r} m_{ik}(r) M_{k\vec{j}}(n-r) + \sum_{k \in \vec{j}} m_{ik}(n) \\ &= \sum_{k \notin \vec{j}} \sum_{r=1}^{n} \binom{n}{r} m_{ik}(r) M_{k\vec{j}}(n-r) + \\ &\quad \sum_{k \notin \vec{j}} p_{ik} M_{k\vec{j}}(n) + \sum_{k \in \vec{j}} m_{ik}(n) \end{aligned} \tag{2}$$

for $i \notin \vec{j}$ and $M_{i\vec{j}}(n) = 0$ for $i \in \vec{j}$, where $p_{ik} = r^*_{ik}(0) \equiv m_{ik}(0)$. The first and third terms on the right hand side will be known prior to an iteration, facilitating a straightforward iteration that solves a set of linear equations at each step.

For a Markov chain with generator matrix Q, Eq. 2 reduces to:

$$-q_{ii} M_{i\vec{j}}(n) = \sum_{k \notin \vec{j}} q_{ik} M_{k\vec{j}}(n) + n M_{i\vec{j}}(n-1) \tag{3}$$

for $i \notin \vec{j}$ and $M_{i\vec{j}}(n) = 0$ for $i \in \vec{j}$. For $n = 0$, we have $M_{i\vec{j}}(0) = 1$ and so each set of moments can be computed iteratively.

3 Generalised Lambda Distributions

3.1 Description

The Generalised Lambda Distribution (GLD) is a family of curves which has ability to assume a wide variety of shapes including the standard distribution types exponential, normal, χ^2, uniform, log-normal etc. Because of this flexibility, GLDs have been extensively used to fit and model continuous probability distributions in diverse application areas such as meteorology, medical trials, financial data modelling and Monte Carlo simulation studies [9].

A GLD is defined as an inverse cumulative distribution (quantile) function $F^{-1}(u)$ (where u takes values between 0 and 1) that yields the value of x such that $F(x) = u$. It has form:

$$F^{-1}(u) \equiv Q_{\lambda_1,\lambda_2,\lambda_3,\lambda_4}(u) \tag{4}$$

where λ_1 the location parameter, λ_2 is the scale parameter and λ_3 and λ_4 are the shape parameters. If λ_3 = λ_4 then the distribution is symmetric. The function Q can take one of two forms, both of which are multi-parameter generalisations of the one-parameter Tukey-Lambda distribution. For notational simplicity in what follows we will omit the λ subscripts and simply write $Q(u)$.

Using the relationships $Q(u) = x$ and $F(x) = u$ and Eq. (4), the probability density function $f(x)$ may be derived as:

$$f(x) = \frac{du}{dx} = \frac{du}{dQ(u)} = \left(\frac{dQ(u)}{du}\right)^{-1} \tag{5}$$

A plot of the density function $f(x)$ can thus be obtained parametrically by plotting $Q(u)$ against $f(Q(u))$ for $0 \leq u \leq 1$.

As we will be fitting the GLD by moments, we note that the kth raw moment of a quantile function $Q(u)$ is:

$$\begin{aligned} E[X^k] &= \int_0^\infty x^k f(x)dx \\ &= \int_0^1 (Q(u))^k \frac{du}{dQ(u)} dQ(u) \\ &= \int_0^1 (Q(u))^k du \end{aligned} \tag{6}$$

3.2 Parameterization

As mentioned, the function Q in Eq. (4) can take on one of two forms. In the original Ramberg-Schmeiser (RS) [16] parameterisation,

$$Q(u) = \lambda_1 + \frac{u^{\lambda_3} - (1-u)^{\lambda_4}}{\lambda_2}$$

However, this parameterisation does not result in a well defined pdf for certain values of λ_3 and λ_4 [15, 9]. This limitation can be partially overcome by introducing Generalised Beta Distributions

(GBDs) to extend the defined area [6, 9]. The later FMKL [7] parameterisation due to Freimer *et al* defines

$$Q(u) = \lambda_1 + \frac{1}{\lambda_2}\left(\frac{u^{\lambda_3}-1}{\lambda_3} - \frac{(1-u)^{\lambda_4}-1}{\lambda_4}\right) \tag{7}$$

which is well defined over the entire λ_3, λ_4 plane. For this reason, we adopt this FMKL parameterisation. Using Eq. (5) and Eq. (7) we have:

$$f(Q(u)) = \frac{\lambda_2}{u^{\lambda_3-1} + (1-u)^{\lambda_4-1}}$$

3.3 Fitting via moment matching

We wish to find GLD parameters $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ such that the mean μ, variance σ^2, skewness α_3 and kurtosis α_4 of the GLD correspond to a given mean $\hat{\mu}$, variance $\hat{\sigma}^2$, skewness $\hat{\alpha_3}$ and kurtosis $\hat{\alpha_4}$. Matching these four measures of distribution is adequate to determine $\lambda_1, \lambda_2, \lambda_3$ and λ_4.

First, we need to determine the central moments of the quantile function $Q(u)$. Eq. (7) can be expanded as [10]:

$$\begin{aligned} Q(u) &= \left(\lambda_1 - \frac{1}{\lambda_2\lambda_3} + \frac{1}{\lambda_2\lambda_4} + \frac{1}{\lambda_2}\left(\frac{u^{\lambda_3}}{\lambda_3} - \frac{(1-u)^{\lambda_4}}{\lambda_4}\right)\right) \\ &= a + bR(u) \end{aligned} \tag{8}$$

where

$$R(u) = \left(\frac{u^{\lambda_3}}{\lambda_3} - \frac{(1-u)^{\lambda_4}}{\lambda_4}\right)$$

Let $\hat{q}_k$ denote the kth central moment of $Q(u)$ and r_k the kth raw moment of $R(u)$. Then, from Eq. (8), the first four central moments of $Q(u)$, can be expressed in terms of the raw moments of $R(u)$ as:

$$\begin{aligned} \hat{q}_1 &= \lambda_1 - 1/(\lambda_2\lambda_3) + 1/(\lambda_2\lambda_4) + r_1/\lambda_2 \\ \hat{q}_2 &= \frac{1}{\lambda_2^2}\left(r_2 - r_1^2\right) \\ \hat{q}_3 &= \frac{1}{\lambda_2^3}\left(r_3 - 3r_1r_2 + 2r_1^3\right) \\ \hat{q}_4 &= \frac{1}{\lambda_2^4}\left(r_4 - 4r_1r_3 + 6r_1^2r_2 - 3r_1^4\right) \end{aligned} \tag{9}$$

From Eq. (6), r_k is given by:

$$r_k = \int_0^1 \left(\frac{u^{\lambda_3}}{\lambda_3} - \frac{(1-u)^{\lambda_4}}{\lambda_4}\right)^k du$$

By binomial expansion on r_k, we have [10]:

$$\begin{aligned} r_k &= \int_0^1 \sum_{j=0}^{k} \binom{k}{j} (-1)^j \frac{u^{\lambda_3(k-j)}}{\lambda_3^{k-j}} \frac{(1-u)^{\lambda_4 j}}{\lambda_4^j} du \\ &= \sum_{j=0}^{k} \frac{(-1)^j}{\lambda_3^{k-j}\lambda_4^j} \binom{k}{j} \beta(\lambda_3(k-j)+1, \lambda_4 j + 1) \end{aligned} \tag{10}$$

where

$$\beta(a,b) = \int_0^1 u^{a-1}(1-u)^{b-1}du$$

The beta function is defined only for positive arguments, so (in common with all moment matching techniques for GLDs) we require

$$\min(\lambda_3, \lambda_4) > -\frac{1}{4}.$$

Now, from Eq. (9) and Eq. (10), it follows that the skewness $\alpha_3 \equiv \hat{q}_3/\hat{q}_2^{3/2}$ and and kurtosis $\alpha_4 \equiv \hat{q}_4/\hat{q}_2^2$ are functions of λ_3 and λ_4 only. By solving the set of two simultaneous non-linear equations

$$\begin{aligned} \alpha_3 &= \hat{\alpha}_3 \\ \alpha_4 &= \hat{\alpha}_4 \end{aligned} \tag{11}$$

we obtain values for λ_3, λ_4. Subsequently λ_2 and λ_1 can be computed as:

$$\begin{aligned} \lambda_2 &= \frac{\sqrt{r_2 - r_1^2}}{\hat{\sigma}} \\ \lambda_1 &= \hat{\mu} + \frac{1}{\lambda_2}\left(\frac{1}{\lambda_3+1} - \frac{1}{\lambda_4+1}\right) \end{aligned}$$

The non-linear equations of Eq. (11) do not have a closed-form solution. However, it is possible to apply numerical methods such as the Nelder-Mead simplex method [11, 13] and Powell's method [12]. Computer software libraries which implement these methods and which perform multi-variable optimization are available; for our results we used Besset's Java library [1]. Corresponding cdfs can be obtained by numerical integration of the pdf (e.g. using the Trapezoidal rule or Simpson's rule).

4 Numerical Results

This section presents results obtained from a variety of Markov and semi-Markov models. In each case, we extract a response time probability density function (pdf) and corresponding cumululative distribution function (cdf) using the exact Laplace transform approach of Section 2.2 [8, 5, 3], and compare it to the approximations calculated using the GLD approach outlined above. We also compare the quality of the results with those provided by the WinMoments tool [14].

4.1 GSPN models

We first apply our technique to two GSPN models of real-life software and hardware systems, namely the Courier model and the Flexible Manufacturing System (FMS) model.

The Courier model is a 45-place GSPN representing the ISO Application, Session and Transport layers of a sliding-window communication protocol (see [17] for full details). The response time of interest is the time taken from the initiation of a transport-layer send to the arrival of the corresponding acknowledgement packet. Fig. 1 shows the exact response time pdf and cdf, calculated in 134 seconds for the $11\,700$ tangible state model using the Laplace-transform based technique of [8], as well as the

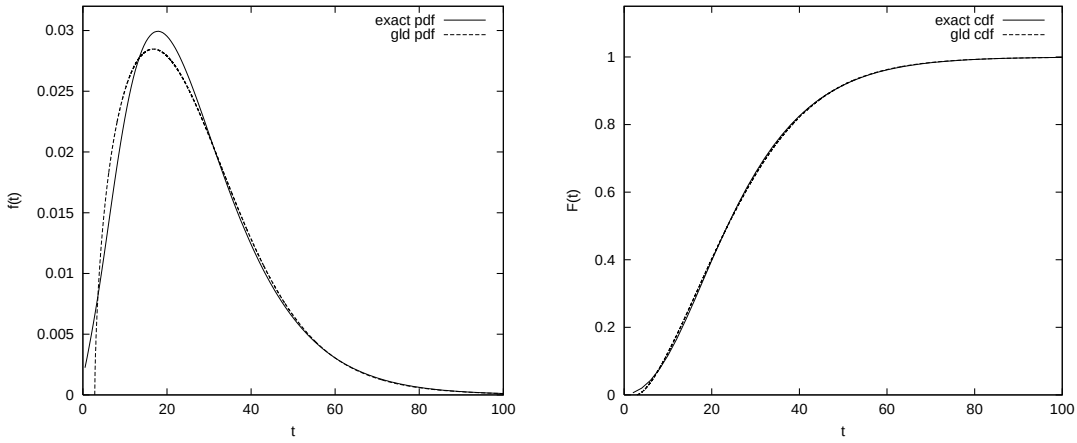


Figure 1: Courier GSPN-model pdf (left) and corresponding cdf (right)

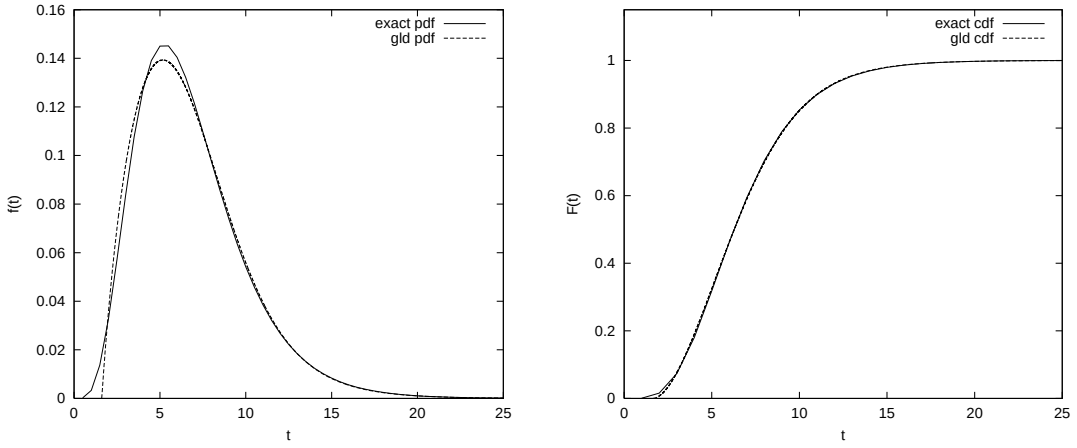


Figure 2: FMS GSPN-model pdf (left) and corresponding cdf (right)

approximate GLD pdf and cdf, calculated in just 0.94 seconds (which includes time taken to calculate moments).

The FMS model is a 22-place GSPN representing an assembly line composed of three types of machines and four types of parts (see [4] for full details). Starting with 4 unprocessed parts of types 1, 2 and 3, we are interested in measuring the time to complete the first processed part of type 4. Fig. 2 shows the exact pdf and cdf, calculated in 834 seconds (13 minutes 54 seconds) for the $35\,910$ tangible state model using the Laplace-transform based technique, as well as the approximate GLD pdf and cdf, calculated in 8.06 seconds.

In both cases, we observe good agreement between the approximate and exact pdfs, and excellent agreement between the approximate and exact cdfs. The latter is particularly useful for accurately estimating response time quantiles.

4.2 Queueing network model

We now apply our method to approximate a cycle time density for a path in a closed tree-like network with 8 customers (see [8] for full details). The cycle time of interest is measured from when a tagged customer arrives at the back of the first queue, and ends when the customer returns to the queue. Fig. 3 shows moderate agreement between the exact and approximate pdfs but excellent agreement between the cdfs.

4.3 Bimodal models

To test the ability of the GLD method to approximate response time densities that are not unimodal, we show results for the cycle time in a branching Erlang model (see [8]). This model is composed of two equiprobable branches, one of which results in an Erlang($3, \lambda_1$) delay, and the other of which results in a Erlang($12, \lambda_2$) delay. Setting $\lambda_1 = \lambda_2 = 1$, we obtain a bimodal density curve, as shown in Fig. 4. The GLD approximation for the pdf does not capture its bimodal nature; however the cdf still shows good agreement. Setting $\lambda_1 = 1$ and $\lambda_2 = 2$, we obtain the almost unimodal curve shown in Fig. 5. The GLD approximation now shows a much better fit (for both pdf and cdf).

4.4 SMSPN models

Moving on to semi-Markov Stochastic Petri net models (see [2] for details of this formalism), we first consider the simple 4-place model shown in Fig. 6. The response time of interest is the time taken for 20 tokens to move from place $p1$ to place $p0$. The exact and approximate pdfs and cdfs for this response time are shown in Fig. 7. The exact distributions are calculated using the iterative algorithm presented in [3].

Figs. 8 and 9 show results for a more complex SMSPN model of a web content authoring service (see [3] for more details). In this system, authors publish content on a number of web servers; there is also a pool of readers who submit requests to the servers to be provided with content. The servers are unreliable and can fail and then recover. A system of 12 writers and 24 readers yields a $15\,257$ state semi-Markov chain.

Fig. 8 shows the distribution of time taken for all writers to commit their updates and all readers to receive their requested content, while Fig. 9 represents the time taken only for the readers to receive their requested content.

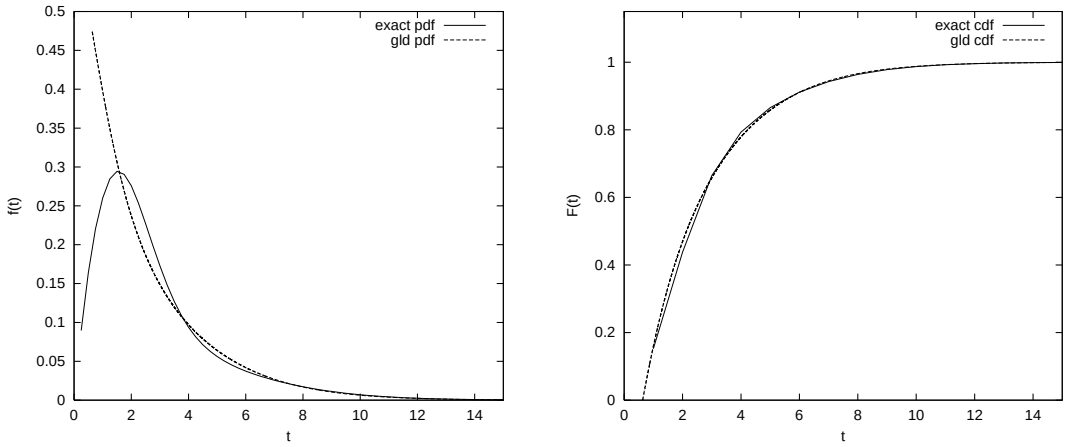


Figure 3: Tree-like queueing network cycle time pdf (left) and corresponding cdf (right)

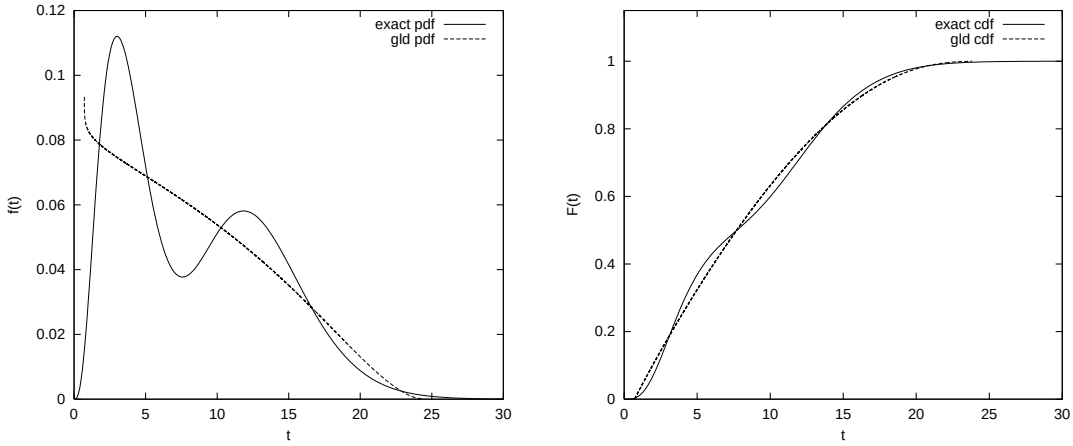


Figure 4: Bimodal branching Erlang pdf (left) and corresponding cdf (right)

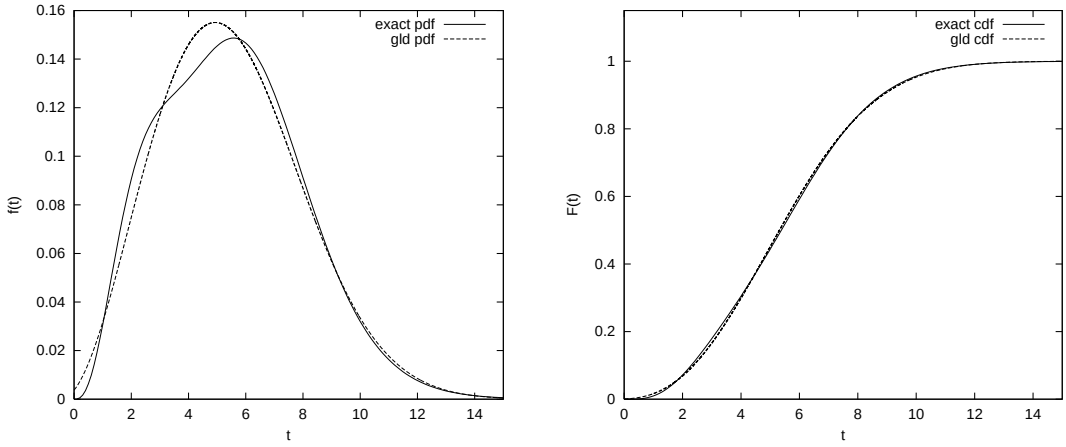


Figure 5: Almost unimodal branching Erlang pdf (left) and corresponding cdf (right)

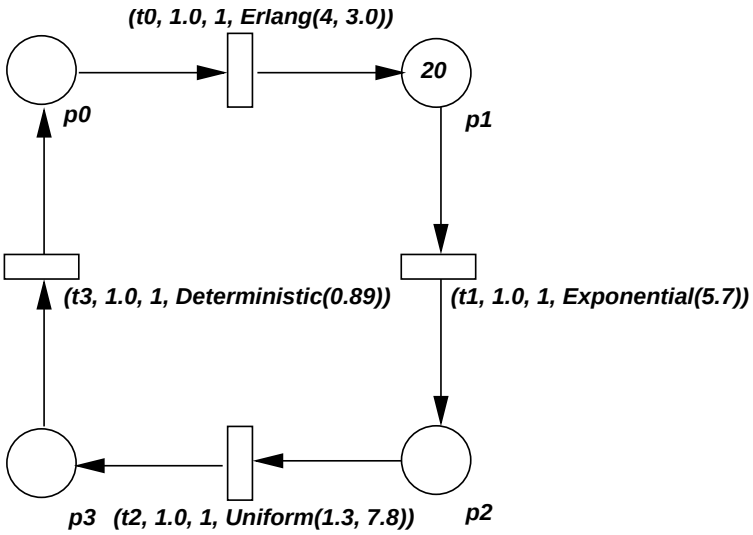


Figure 6: Simple 4-place semi-Markov SPN model.

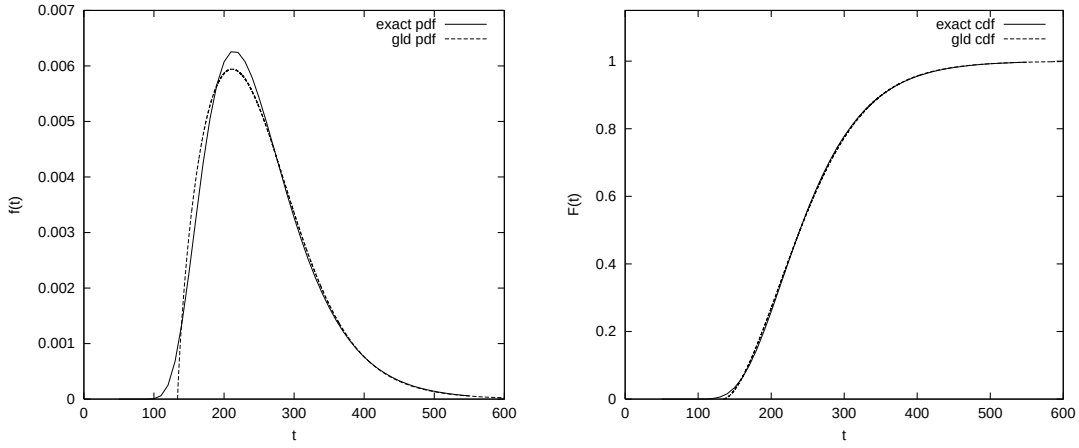


Figure 7: Simple 4-place semi-Markov model pdf (left) and corresponding cdf (right)

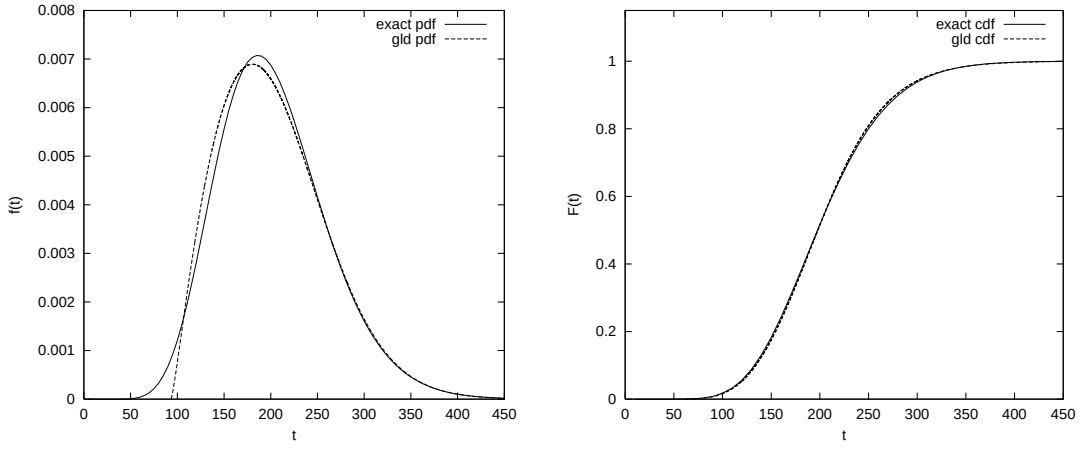


Figure 8: Webserver SMSPN-model Response Time 1 pdf (left) and corresponding cdf (right)

4.5 Comparison with WinMoments tool

Finally, Fig. 10 compares the cdf approximations produced by the GLD method with the upper and lower bounds computed by Rácz's WinMoments tool [14] for the Webserver Response Time 2 cdf and the first branching Erlang cdf. Given a finite number of moments and a t point, WinMoments calculates upper and lower bounds on the value of $F(t)$. In both cases, the GLD approximation lies well within the WinMoment-calculated bounds, and provides a better approximation to the actual cdf than the mid-point of the bounds. Note that we have only used the first four moments as input to WinMoments; given a higher number of moments we would expect the bounds to be narrower.

5 Conclusions

We have conducted a study into the rapid approximation of response time densities and quantiles in Markov and semi-Markov models using Generalised Lambda Distributions. The results presented demonstrate that this method provides a good estimation of pdfs and excellent estimation of cdfs from the first four moments of response time, most notably in the cases where the response time densities are unimodal. The approximations produced compare favourably with the bounds generated by the WinMoments tool.

The GLD-based estimation technique offers offers significant predictive insights at low cost when compared to an exact Laplace transform-based approach. The exact technique requires the solution of a large number of systems of linear equations (typically greater than 400), the complexity of which is a function of the number of states in the stochastic model. The approximation technique presented here, however, requires the solution of only four sets of these linear equations in order to calculate the first four moments of the response time distribution, plus the time taken to use these moments to perform the estimation (which is independent of the number of states).

As future work, we intend to investigate the sensitivity of the GLD method to perturbations in the moments and to seek an even wider range of application examples. For bimodal densities, it may be possible to improve the accuracy of the GLD method by considering them as the superposition of two unimodal densities.

6 Acknowledgements

The authors would like to thank Pete Harrison for his helpful suggestions and comments.

References

[1] D.H. Besset. *Object-Orientated Implementation of Numerical Methods: An Introduction with Java and Smalltalk.* Morgan Kaufmann, 2001.

[2] J. T. Bradley, N. J. Dingle, W. J. Knottenbelt, and P. G. Harrison. Performance queries on semi-Markov stochastic Petri nets with an extended Continuous Stochastic Logic. In *Proceedings of Petri Nets and Performance Models (PNPM'03)*, Urbana-Champaign, IL, September 2003. (to appear).

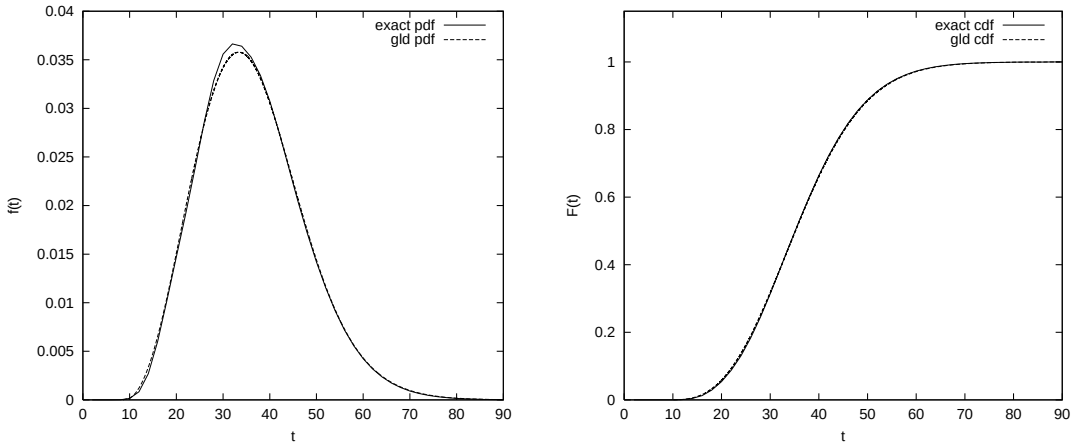


Figure 9: Webserver SMSPN-model Response Time 2 pdf (left) and corresponding cdf (right)

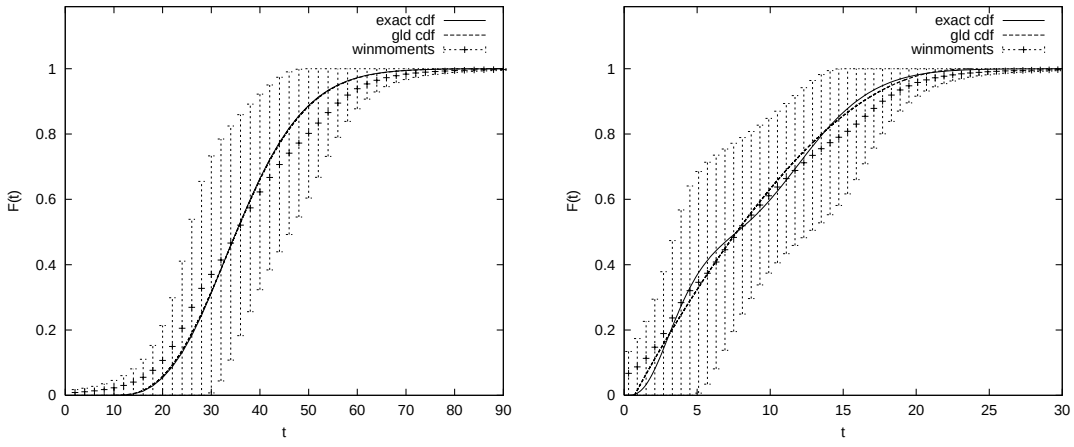


Figure 10: Webserver SMSPN-model Response Time 2 cdf (left) and first branching Erlang cdf (right) compared with the bounds generated by the WinMoments tool.

[3] J.T. Bradley, N.J. Dingle, W.J. Knottenbelt, and H.J. Wilson. Hypergraph-based parallel computation of passage time densities in large semi-markov models. In *Proc. 4th International Meeting on the Numerical Solution of Markov Chains (NSMC '03)*, pages 99–120, Urbana-Champaign, Illinois, September 2003.

[4] G. Ciardo and K.S. Trivedi. A decomposition approach for stochastic reward net models. *Performance Evaluation*, 18(1):37–59, 1993.

[5] N.J. Dingle, P.G. Harrison, and W.J. Knottenbelt. Response time densities in Generalised Stochastic Petri Net models. In *Proceedings of the 3rd International Workshop on Software and Performance (WOSP 2002)*, pages 46–54, Rome, July 24th–26th 2002.

[6] E.J. Dudewicz and Z.A. Karian. The Extended Generalized Lambda Distribution (EGLD) system for fitting distributions to data with moments II: Tables. *American Journal of Mathematical and Management Sciences*, 16(3 & 4):271–330, 1996.

[7] M. Freimer, G. Mudholkar, G. Kollia, and C. Lin. A study of the generalized Tukey Lambda family. *Communications in Statistics: Theory and Methods*, 17(10):3547–3567, 1988.

[8] P.G. Harrison and W.J. Knottenbelt. Passage time distributions in large Markov chains. In *Proc. ACM SIGMETRICS 2002*, Marina Del Rey, California, June 2002.

[9] Z. Karian and E. Dudewicz. *Fitting Statistical Distributions: The Generalized Lambda Distribution and Generalized Bootstrap Methods*. CRC Press, Boca Raton, 2000.

[10] A. Lakhany and H. Mausser. Estimating the parameters of the Generalized Lambda Distribution. *Algo Research Quarterly*, 3(3):47–58, 2000.

[11] J.A. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7:308–313, 1965.

[12] M.J.D. Powell. An iterative method for finding stationary values of a function of several variables. *Computer Journal*, 5(2):147–151, 1962.

[13] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 2nd edition, 1993.

[14] S. Rácz. *Numerical Analysis of Communication Systems through Markov Reward Models*. PhD thesis, Budapest University of Technology and Economics, 2002.

[15] J.S. Ramberg, E. Dudewicz, P. Tadikamalla, and E. Mykytka. A probability distribution and its uses in fitting data. *Technometrics*, 21(2):201–214, 1979.

[16] J.S. Ramberg and B.W. Schmeiser. An approximate method for generating asymmetric random variables. *Communications of the ACM*, 17(2):78–82, 1974.

[17] C.M. Woodside and Y. Li. Performance Petri net analysis of communication protocol software by delay-equivalent aggregation. In *Proceedings of the 4th International Workshop on Petri nets and Performance Models*, pages 64–73, Melbourne, Australia, 2–5 December 1991. IEEE Computer Society Press.