

Fast Fitting of Convolutions Using Rational Approximations

Denis Cousineau (denis.cousineau@umontreal.ca)

Université de Montréal, C.P. 6128, succ. Centre-ville
Montréal (Québec), H3C 3J7 Canada

Abstract

Models of response time distributions are powerful models because they capture mean response times as well as variability and asymmetry. Some of these models are expressed with a closed-form equation (e.g. the Weibull distribution). However, a large number of others are not available in closed form, including convolutions. Fitting such distributions with the likelihood method is still possible with the help of numerical integration techniques. However, the time required to fit a single subject is a matter of days. We present an alternative to numerical integration based on rational approximations. This approach is 100 times faster so that fitting a single subject can be done in less than an hour. As shown with simulations, the approach can be fully automatized, and is both reliable and accurate.

Introduction

Response times (RT) give very precious information on elementary mental processes. Psychological models can be used to fit mean RT alone, or means and standard deviations of RT simultaneously (Cousineau & Larochelle, in press). However, a more stringent test is imposed when a model must fit the whole distribution of RT (Cousineau & Shiffrin, in press). Some models predict simple distributions and therefore are easy to fit to the RT data using the maximum likelihood method (described next). This is the case for the diffusion models which predict a Wald distribution (Ratcliff, Van Zandt, & McKoon, 1999). However, such models are the exceptions: a large number of models predict that the observed RT distributions will conform to a convolution of two or more simple distributions which cannot be simplified into a closed-form equation.

A simple psychological model that predicts a convolution is one which assumes two stages, for example one stage for encoding the stimulus and one stage for making a motor response. The observed RT is the sum of the two processing times. In this situation, the mean RT is the sum of the two mean processing times. The variance in RT is also the sum of the variance of the two processes. However, the RT distribution is **not** the sum of the two processes distributions. It is obtained by an operation called a convolution of the two processes distributions.

Because two-stage models are very frequent,

convolutions occur naturally in psychology.

Convolutions

A convolution is the distribution of the sum of two or more random values. Formally, let T be a random deviate and T_1 and T_2 be the processes that contribute to T , such that $T = T_1 + T_2$. Often, only T is observable. We can estimate the probability that on a given trial, T will take the value t with the equation:

$$\Pr(T = t) = \Pr(T_1 = s \& T_2 = t - s)$$

for all s (because s and $t - s$ cannot be negative response times in psychology, we have the constraint that $0 < s < t$). Assuming that the second processing time is independent of the first, we can write:

$$\begin{aligned} \Pr(T = t) &= \Pr(T_1 = s) \times \Pr(T_2 = t - s) \forall s \\ &= \int_0^t \Pr(T_1 = s) \times \Pr(T_2 = t - s) ds \end{aligned} \quad (1)$$

Equation 1 is the basic equation for convolutions. When assumptions are given on the distributions of T_1 and T_2 , Eq. 1 can sometimes be solved in closed form. Let $f_1(t | \theta_1)$ and $f_2(t | \theta_2)$ be two probability density functions (pdf) for T_1 and T_2 with parameters θ_1 and θ_2 respectively. The convolution is often denoted using the * so that $(f_1 * f_2)(t)$ yields the density that the sum of the two processes equals t .

As an example, if f_1 is the normal distribution with parameters $\{\mu_1, \sigma_1^2\}$ and f_2 is also normal with parameters $\{\mu_2, \sigma_2^2\}$, Eq. 1 is easy to solve, and the result of $(f_1 * f_2)(t)$ is normal with parameters $\{\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2\}$ (Cramér, 1947).

Another example often seen in psychology relates to the Poisson race model (Townsend & Ashby, 1983). This model assumes that when n spikes reach a certain neuron, this neuron is triggered. Suppose that under given conditions, a spike occurs every τ milliseconds on average. If the time between two spikes is random following a Poisson process (an exponential distribution), then the total time before the neuron fires is given by $T = T_1 + T_2 + \dots + T_n$. The distribution of T is a convolution of n exponential distributions. Again, this convolution can be solved and yields a gamma distribution with two parameters $\{n, \tau\}$ (Luce, 1986).

On other occasions, the convolution cannot be solved in closed form but good approximations exist for

its computation. This is the case for the exGaussian distribution (Ratcliff, 1979). It is based on the assumptions that the decision process is modeled by an exponential distribution and that all the other processes, owing to the theory of errors (Gauss, 1809/1864), are normally distributed. The observed time is therefore a sum of a normal component and an exponential component. Using Eq. 1, we find:

$$\begin{aligned} \Pr(\mathbf{T} = t) &= \int_0^t \frac{e^{-\frac{(s-\mu)^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma} \times \frac{e^{-\frac{(t-s)}{\tau}}}{\tau} ds \\ &= \frac{1}{2\tau} e^{\frac{\mu-RT}{\tau} + \frac{\sigma^2}{2\tau^2}} \left(1 + \operatorname{erf} \left(\frac{RT - \mu}{\sigma} - \frac{\sigma}{\tau} \right) \right) \end{aligned}$$

where $\operatorname{erf}(t)$, often called the error function, is given by

$$\operatorname{erf}(t) = \int_0^t e^{-\frac{s^2}{2}} ds$$

This integral is not available in closed form. However, good approximations exist and are quite efficient in terms of computation time. The most commonly used approximation was obtained by using a ratio of two cubic polynomials (Kennedy and Gentle, 1980). Sometimes, by subdividing the curve in sub intervals, better approximations can be found. The erf function implemented on most programming platform is subdivided in three intervals, and for each, a rational approximation was found. As an example, the following is the rational approximation for the $\operatorname{erf}(x)$ function within 0 and 0.5:

$$\frac{3.2027 \times 10^{-7} + 1.1283x + 34.011x^2 - 1.0401x^3}{1 + 30.140x - 0.56796x^2 + 9.9358x^3}$$

When no approximation is available, an equation containing an integral can still be evaluated using numerical integration techniques (mostly based on Monte Carlo and Markov chain techniques). To estimate $(f_1 * f_2)(t)$ at a single point t , these techniques use an approach analogous to throwing a large number of darts and finding the proportion of them that are below the curve. Hence, to evaluate the function at one point, hundreds or thousands of computations have to be done, resulting in very slow computations.

This paper aims at presenting a solution to speed up fitting of convolutions. When fitting a convolution to RT data, the function has to be estimated for each datum and each parameter set explored. Because the convolution is so slow to estimate, finding the best-fitting parameters typically requires from two to five days per subject on a 2.0 GHz PC. The approach described next will reduce this time by a factor of 100, so that fitting one subject is possible in about an hour.

Distribution Fitting

Before presenting the approach, we briefly describe the maximum likelihood method used to find best-fitting parameters (Cousineau & Larochelle, 1997). It consists in finding the parameters that make the data the most likely. Let the likelihood L of the data $\{t_i, i = 1 \dots n\}$ given parameters θ be:

$$L(t_i, \theta) = \Pr(\mathbf{T} = t_1 \ \& \ \mathbf{T} = t_2 \ \& \dots \ \& \ \mathbf{T} = t_n)$$

where n is the number of data to be fitted. When the data are independent, we can simplify the equation to:

$$\begin{aligned} L(t_i, \theta) &= \prod_{i=1}^n \Pr(\mathbf{T} = t_i) \\ &= \prod_{i=1}^n f(t_i | \theta) \end{aligned}$$

using f to denote the pdf underlying the data which is a function of the parameters θ . The best-fitting parameters are those that make the overall probability L closest to 1. To avoid underflow, it is often more convenient to use the log of the probabilities, so that the log likelihood LL is given by:

$$\begin{aligned} LL(t_i, \theta) &= \log \left(\prod_{i=1}^n f(t_i | \theta) \right) \\ &= \sum_{i=1}^n \log(f(t_i | \theta)) \end{aligned} \quad (2)$$

The best-fitting parameters θ^* are those that maximize $LL(t_i, \theta)$ or equivalently, that minimize minus $LL(t_i, \theta)$:

$$\operatorname{Max}_{\theta \in \Theta} LL(t_i, \theta)$$

Maximizing a function is an iterative process that requires considering many possible parameters. Some methods are based on gradient descents (Chandler, 1965), others are based on geometric methods (simplex, Nelder and Mead, 1965). Typical code in *Mathematica* is given in Listing 1.

Note that the programs in Listings 1 and 2 are not optimized for speed. By using Apply and Map instead of the summation, the code will operate much faster:

```
-Apply[Plus, Map[Log[f[#, {p1, p2, p3}]] &, data]]
```

In the simulations reported next, the code was optimized for speed.

Objectives

Distribution fitting of a convolution can be accomplished using numerical integration. However, this approach is too slow to be useful in any practical application. We propose in this paper to systematize the use of approximations. There exists nowadays algorithms that can find a good approximation to any function within a given interval $[a, b]$ by evaluating this

```

<< Statistics`ContinuousDistributions`
data = ReadList[file, Real];
n = Length[data];
f[x_, θ_] := PDF[WeibullDistribution[θ[[3]], θ[[2]], x - θ[[1]]] /; x ≥ θ[[1]]
f[x_, θ_] := 10-50 /; x < θ[[1]]
FindMinimum[
  - ∑i=1n Log[f[data[[i]], {p1, p2, p3}]],
  {p1, 250, 300},
  {p2, 50, 150},
  {p3, 1.8, 2.2}
]

```

Listing 1. A typical Mathematica code for finding the best-fitting parameters. The distribution f in the example is a Weibull distribution (built-in) with three parameters $\theta[[1]]$ (the position), $\theta[[2]]$ (the scale) and $\theta[[3]]$ (the shape).

function at only a few selected positions. As soon as the data set is sufficiently large ($n > n_0$), it is faster to find an approximation and use it n times that to estimate the original function n times using numerical integration. In the following, we explore one type of approximation, the rational approximation. We will examine how reliable the process is, how accurate the resulting approximations are, and how much faster fitting using approximations is.

Finding Rational Approximations

A rational approximation to an equation $f(x)$ in the interval $[a, b]$ is another equation composed by the ratio of two polynomials of degree m and n (m can be equal to n). Let $r_{m,n}(x)$ denote such an approximation:

$$r_{m,n}(x) = \frac{a_0 + a_1x^1 + \dots + a_mx^m}{b_0 + b_1x^1 + \dots + b_nx^n}$$

in which the coefficients a_0, \dots, a_m and $b_0, \dots, b_n \in \mathbb{R}$ have to be chosen so that the error

$$\varepsilon(x) = |f(x) - r_{m,n}(x)|$$

is small (Jackson, 1930). If f depends on parameters θ , every time θ are changed, a new approximation has to be found.

It is possible to demonstrate (Petrushev & Popov, 1987) that for any continuous function $f(x)$, x being restricted to a domain $[a, b]$, there exists a single set of coefficients $\{a_0, \dots, a_m, b_0, \dots, b_n\}$ so that the difference is the smallest. Further, there exists an algorithm, the Remez algorithm (described in Petrushev and Popov, 1987) that can find these coefficients rapidly (the algorithm is said to be quadratically convergent).

The Remez algorithm is not particularly complex to implement and is already provided with some software (such as *Mathematica*, by loading the *NumericalMath`Approximations`* library). Fitting a distribution is therefore a two-step process: For a tested θ , find an approximation, then compute LL using the approximation instead of the original equation. Listing 2 shows an example of code in *Mathematica* which automatically finds a rational approximation $r(x)$ of degree 10, 10.

Reliability of the Approach

We tested the approach under three aspects: its reliability, its accuracy and its speed. The following simulations were run using Weibull distributions. It was used because of its relations to psychological models (Cousineau, Goodman & Shiffrin, 2002). Its pdf is given by:

$$f(x | \alpha, \beta, \gamma) = \gamma\beta^{-\gamma} (x - \alpha)^{\gamma-1} e^{-\left(\frac{x-\alpha}{\beta}\right)^\gamma}$$

where α is the shift parameter, β is the scale parameter, and γ is the shape parameter. When the shape is 1, the Weibull reduces to a shifted exponential distribution. Unless the shape parameter is 1, a convolution of two Weibull distributions never yields a closed-form equation.

Reliability

Reliability measures whether the approach always returns an approximation. To test this, we generated random parameters for two Weibull components that were convolved. We then looked whether an

```

<< Statistics`ContinuousDistributions`
<< NumericalMath`Approximations`
data = ReadList[file, Real];
n = Length[data];
f[x_, θ_] := PDF[WeibullDistribution [θ[[3]], θ[[2]], x - θ[[1]]] /; x ≥ θ[[1]]
f[x_, θ_] := 10-50 /; x < θ[[1]]
FindMinimum[
  r[x_] = RationalInterpolation[f[x, {p1, p2, p3}], {x, 10, 10},
    {x, Min[data], Max[data]}];
  -∑i=1n Log[If[r[data[[i]]] ≤ 0, 10-16, r[data[[i]]] ]],

  {p1, 250, 300},
  {p2, 50, 150},
  {p3, 1.8, 2.2}
]

```

Listing 2. A typical Mathematica code for finding the best-fitting parameters using an approximation. The approximation is found within the interval $[\text{Min}(t_i), \text{Max}(t_i)]$

approximation was found.

The parameters were as follow: α_1 and α_2 were normal with mean 250 and standard deviation 40; β_1 and β_2 were normal with mean 60 and standard deviation 10; γ_1 and γ_2 were normal with mean 2 and standard deviation 0.2. The means were chosen so that the resulting distributions look like an RT distribution. $\alpha_1 + \alpha_2$ is a lower bound so that $\Pr(T < \alpha_1 + \alpha_2) = 0$.

The approximations requested in all the following simulations were ratios of polynomials of degree 10 on both the numerator and the denominator. The approximation had to be valid within the interval $[\alpha_1 + \alpha_2, \alpha_1 + \alpha_2 + 6(\beta_1 + \beta_2)]$, going well beyond four standard deviation above the mean.

Repeated over a thousand replications, the process never failed, always returning an approximation. Figure 1 shows one such approximation. As seen the approximation is visually quite good. However, it is sometimes hovering around zero in the tails. The curve should not go beyond zero since it is suppose to approximate probabilities. Further, later on, when we proceed to distribution fitting, computing the log of a negative value stops the minimization algorithm. To avoid these problems, we added a condition that if the approximation returns a value below zero, 10^{-16} should be returned instead (see Listing 2). Note that the true probability of sampling a RT in these areas is quite small (once every million samples approximately).

Accuracy

Accuracy being crucial, we tested it from three different points of view.

Area Under the Curve The maximum likelihood method assumes that a probability density function is used in Eq. 2. Because the area under a pdf is 1, the area under the approximation should also be 1. We generated the parameters of a convolution randomly as above and then generated an approximation $r(x)$ over the interval $[\alpha_1 + \alpha_2, \alpha_1 + \alpha_2 + 6(\beta_1 + \beta_2)]$. Afterward, the area under the curve was estimated.

Repeated over a thousand replications, the largest absolute deviation to 1 was in the order of 10^{-10} which is of the same magnitude as the numerical integration error.

Likelihood Result The fitting procedure relies on the $LL(t_i, \theta)$ quantity to guide the search for the optimal parameters. Therefore, this quantity must be very accurate. To test this, we first generated random parameters as above, followed by the generation of a sample containing 1200 deviates. One deviate was obtained by generating two Weibull deviates with parameters $\{\alpha_1, \beta_1, \gamma_1\}$ and $\{\alpha_2, \beta_2, \gamma_2\}$ respectively and adding them. Finally, the quantity LL was computed using the true parameters, once with the convolution equation (and numerical integration), and once with the rational approximation. The rational approximation was

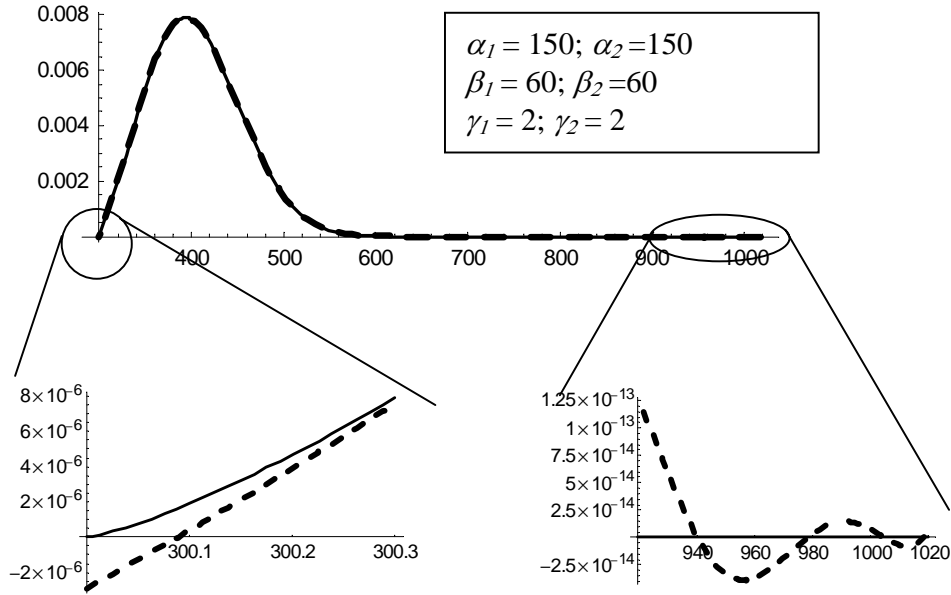


Figure 1. An example of approximation with magnification of the tails. The full line represents the true function $(f_1 * f_2)(t)$ and the dash line represents its approximation $r(t)$

computed over the interval $[\text{Min}(\text{sample}), \text{Max}(\text{sample})]$.

Repeated over a thousand replications, the log likelihood using the convolution was on average -6299.566. The absolute deviation between the LL of the convolution and the LL of the approximation was on average 0.011, that is, an error in the order of 10^{-6} .

Parameter Estimation We also checked the error in the best-fitting parameter estimated. Because fitting parameters using a convolution is so slow (see next), we used a single Weibull distribution instead of a convolution. We generated random parameters α, β, γ in the same manner as previously, followed by the generation of a sample containing 1200 deviates. The best-fitting parameters of the sample were first found using the true Weibull pdf equation, returning θ_f^* and second, using the rational approximation to it, returning θ_r^* . To assess the difference between the two sets of parameters, we looked at the Euclidian distance between them:

$$\|\theta_f^* - \theta_r^*\|$$

Repeated over a thousand replications, the average distance between the two best-fitting solutions was 0.00014, an error in the order of 10^{-8} .

The efficiency of the likelihood method was

explored in Cousineau, Brown & Heathcote (in press). They found that the error between parameters from two samples generated with the same parameters was on average 0.2. This is larger than the error between the two methods as reported here by a factor of about 10^2 .

The second and third tests demonstrate that the error arising from using a rational approximation is totally negligible, being smaller than the error that would occur if numerical integration was used, and being much smaller than the sampling error.

At this point, the two approaches are equivalent. They will diverge on the time they require.

Computation Times

To assess computation times, we generated random parameters for a convolution as above and from these, a sample containing 1200 deviates. Then, we measured the time taken to compute LL once with the true parameters and the convolution equation. We did not try to find the best-fitting parameters because the minimization process needs to compute LL from a hundred to a thousand times to converge onto the optimal parameters, slowing down the simulations accordingly. Using the same parameters and the same sample, we also generated an approximation $r(x)$ to the convolution and computed LL using the approximation. We recorded the time taken to find the approximation

and the time taken to compute LL with the approximation.

Repeated over a thousand replications, the average time to compute LL using the convolution required 164 seconds on a 2.0 GHz PC with *Mathematica* 4.1. The times varied between 60 and 600 seconds.

As for the second method, the average time to find an approximation $r(x)$ was 0.96 second (varying from 0.50 to 1.60 seconds). The variations in the time to find an approximation came from the Remez algorithm requiring one, two or three iterations to find the best rational approximation. The average time to compute LL with the approximation was nearly constant at 0.59 seconds (the variations came from the operating system, Windows 2000).

On average, the two-step method (finding an approximation, using it 1200 times) is 106 times faster than using a convolution 1200 times.

Of course, *Mathematica* is notably known to be slow (but see version 5). Using, for example, *Matlab* would speed-up these computations by about a factor of 20. However, all the computations would equally benefit from this general speed-up so that the 100 to 1 ratio in favor of using an approximation would still be present.

Conclusion

The method presented here relies on brut computational power to find an approximation every time a new parameter set is tested. It does speed up computation times considerably. In fact, we can estimate from the above that as soon as the sample size is 8 or more, it is preferable to use rational approximations. In addition, the whole process can be automatized using the Remez algorithm which is already implemented in *Mathematica*.

Yet, the process might be made even faster. For example, the exGaussian distribution is based on an approximation. However, it was possible to isolate the approximation from the parameters so that the same approximation (erf) is always used whatever the parameters θ . As such, the initial time to find the approximation occurred only once, and is no longer part of the computation. In the above simulations, it would eliminate the 0.96 second interval to find the approximation, resulting in a near 300 times speed-up in computation. It remains to be demonstrated whether convolutions (of two Weibull distributions for example) could be reduced to a single approximation independent of the parameters.

Acknowledgments

This research was supported in part by the Fonds pour la formation de Chercheurs et l'Aide à la Recherche and

the Conseil de Recherches en Sciences Naturelles et en Génie du Canada.

References

- Chandler, P. J. (1965). Subroutine STEPIT: An algorithm that finds the values of the parameters which minimize a given continuous function [computer program]. Bloomington: Indiana University, Quantum chemistry.
- Cousineau, D. Larochelle, S. (1997). *PASTIS: A Program for Curve and Distribution Analyses*. Behavior Research Methods, Instruments, & Computers, 29, 542-548.
- Cousineau, D. Larochelle, S. (in press). *Visual-Memory search: An integrative perspective*. Psychological Research.
- Cousineau, D., Brown, S., Heathcote, A. (in press). *Fitting distributions using maximum likelihood: Methods and packages*. Behavior Research Methods, Instruments, & Computers.
- Cousineau, D., Goodman, V. Shiffrin, R. M. (2002). *Extending statistics of extremes to distributions varying on position and scale, and implication for race models*. Journal of Mathematical Psychology, 46, 431-454.
- Cousineau, D., Shiffrin, R. M. (in press). *Termination of a visual search with large display size effect*. Spatial Vision.
- Cramér, H. (1946). Mathematical Methods of Statistics. Princeton: Princeton University Press.
- Gauss, C. F. (1809/1864). Theoria Motus Corporum Coelestium in Sectionibus Conicis Solem Ambientum. Paris: A. Bertrand.
- Jackson, D. (1930). The theory of approximation. New York: American Mathematical Society.
- Kennedy, W. J. Gentle, J. E. (1980). Statistical computing. New York: Marcel Dekker inc.
- Luce, R. D. (1986). Response times, their role in inferring elementary mental organization. New York: Oxford University Press.
- Nelder, J. A. Mead, R. (1965). *A simplex method for function minimization*. The computer journal, 7, 308-313.
- Petrushev, P. P., Popov, V. A. (1987). Rational approximation of real functions. Cambridge: Cambridge University Press.
- Ratcliff, R. (1979). *Group Reaction Time Distributions and an Analysis of Distribution Statistics*. Psychological Bulletin, 86, 446-461.
- Ratcliff, R., Van Zandt, T. McKoon, G. (1999). *Connectionist and diffusion models of reaction time*. Psychological Review, 106, 261-300.
- Townsend, J. T. Ashby, F. G. (1983). Stochastic Modeling of Elementary Psychological Processes. Cambridge, England: Cambridge University Press.