Technical Manual for QMPE v2.18: Fortran code to fit Response Time Distributions

3 authors, including:

Scott Brown
University of Newcastle
127 PUBLICATIONS **5,815** CITATIONS

SEE PROFILE

Andrew Heathcote
University of Tasmania
265 PUBLICATIONS **5,598** CITATIONS

SEE PROFILE

Technical Manual for QMPE v2.18:

Fortran code to fit Response Time Distributions

© Scott Brown, Denis Cousinau
and Andrew Heathcote

Introduction

Estimating distributional parameters from empirical data is a decidedly non-trival task, especially given the conditions which usually prevail in psychological measurement: high levels of noise and small sample sizes. Various estimation methods have been proposed, most of which fall into two classes: least-squares estimation based on statistics calculated from the theoretical and empirical distributions and maximum likelihood estimation. More recently Heathcote, Brown and Mewhort (in press) suggested the use of a new variant of maximum likelihood estimation: quantile maximum likelihood (QMP). This method was shown to have both smaller bias and variance than Van Zandt's (2000) best method: continuous maximum likelihood (CML). QMP fitting was shown to perform much more effectively in small sample sizes: for example, CML typically requires at least 100 sample data points, but QMP seems to give good results with sample sizes as small as 40. This represents a major and very useful advance in methodology, given that RT sample sizes are often limited to well below 100 points by experimental considerations.

The most important disadvantages of QMP fitting are that it is more complex to implement than other methods, and is more computationally intensive to use. To remove these problems, and to take the burdens of calculating gradients and Hessians for the likelihood objective function from the end user, we present an open-source-code program for fitting the ex-Gaussian distribution by QMP and CML methods.

QMP Fitting

Heathcote et al. (in press) suggested a new method of maximum likelihood fitting, QMP, which is applicable to any distribution function. Standard continuous maximum likelihood fitting (CML) is applied to raw data and works by maximising the likelihood of the distribution's parameters given the data. When one assumes that the data are identical and independent samples from a fixed pdf, $\underline{f}(\underline{t},\theta)$, then the likelihood of a sample of data $\{x_1,\ldots, x_N\}$ is given by the product of the densities at each data point:

$$L(x_1, x_2,..., x_N \mid \theta) \propto \prod_{i=1}^{N} f(x_i, \theta) \qquad (1)$$

QMP is a another form of maximum likelihood estimation where weaker assumptions are made on the model that underlies the data. QMP estimation begins with the specification of a set of probabilities, $\{\underline{p}_i: i=0\ldots\underline{m}\}$, where $\underline{p}_1=0$, $\underline{p}_i<\underline{p}_{i+1}$ and $\underline{p}_m=1$, such that $p(t<\underline{q}_i)=\underline{p}_i$, where the set $\{\underline{q}_i: i=0\ldots\underline{m}\}$ are quantiles. QMP estimation then assumes that the statistical model for the data – whatever that may be – is such that the frequency of observations in the inter-quantile ranges is equal to that which would be observed if data were generated by the CML model. This model leads to the likelihood function given as Equation 3.

$$L(x_1, x_2,..., x_{N_q} \mid \theta) \propto \prod_{i=1}^{N_q+1} \left(\int_{q_{i-1}}^{q_i} f(t, \theta) dt \right)^{n_i} \qquad (2)$$

Where n_i is the number of observations observed between the (i-1)th and i-th quantiles (for convenience of notation, we have defined $q_0=-\infty$ and $q_{m+1}=+\infty$). Maximising this quantity over the parameter vector θ is what we refer to as QMP estimation.

A comparison of Equations 1 and 2 reveals why QMP estimation is more computationally complex than CML. Whereas CML (Equation 1) requires only as many evaluations of $\underline{f}(\underline{t},\theta)$ as there are data points QMP requires m+1 integrations over $\underline{f}(t,\theta)$. This is computationally trivial in the case where $\underline{f}(t,\theta)$ has a closed-form for it's cumulative distribution function (CDF). Unfortunately, for many practical distributions, and for the ex-Gaussian distribution considered below, such an expression either does not exist or is inaccurate, and so numerical quadrature must be used in it's place making QMP fitting typically many orders of magnitude more computationally expensive.

The QMPE Program – An Overview

Given data in a reasonably standard format (see below) QMPE will work on each "cell" of a data file separately, where a cell represents a complete sample of N points from a fixed distribution. QMPE will then calculate any required statistics from this sample. If using CML fitting, no calculation is required. If using QMP fitting, quantiles are estimated from the sample. Any number of quantiles (up to N-1) may be estimated, for any user-specified values of quantile points (see Technical Considerations below for a discussion of how data runs are processed). The calculated sample statistics are then used as the data set for the likelihood maximisation procedure. A start point for this search is generated based on the estimation of moments from the data (see Heathcote, 1996) and search proceeds using a conjugate gradient algorithm (we based our routines on those of Press, Teukolsky, Vetterling & Flannery, 1992, but made several changes for robustness and computational efficiency). The gradient for the likelihood function is evaluated analytically at each point – see Appendices A and B for expressions. Convergence is defined by the fulfilment of either of two conditions: a sufficiently small proportional change in likelihood, or a sufficiently small proportional change in the estimated parameters.

Once the maximum likelihood parameters have been estimated, several details of the best-fitting distribution are determined. Firstly, the Hessian of the negative log likelihood function is evaluated. Calculating analytic expressions for this matrix was another arduous task, so these expressions are included in Appendices A and B, along with some details of their derivation which may aid similar derivations for pdfs other than the those already implemented (ex-Gaussian, Weibull, shifted log-normal, shifted Wald, Gumbel, Gamma). The Hessian matrix is inverted to produce an estimate of the variance-covariance matrix of the maximum likelihood parameter estimates, and this matrix is scaled to give estimates of parameters' standard errors and correlations. Note that this method of estimating the variance-covariance matrix is only valid when the log-likelihood surface in the region of the maximum is closely approximated by a quadratic form (Bates & Watts, 1988). The log-likelihood function may not satisfy this condition, and hence the standard error and correlation estimates may be untrustworthy, in some instances. For example, when a parameter estimate is close to zero, the true confidence regions may be vastly different from those estimated using the standard errors returned by QMPE.

Finally, the expected values (quantiles if using QMP or raw data values for CML) for the maximum likelihood solution are calculated so that they can be compared with the statistics estimated from data, and systematic deviation detected. The expected statistics are calculated by line search along the numerically approximated CDF. The line search algorithm used is a hybrid that combines the extreme efficiency of Newton's method, where it is stable, with the robustness of the bisection method, where Newton's method is unstable. An exit code is then determined which summarises the state of the fit, containing important information such as convergence properties and Hessian singularity. Finally, the parameters of the maximum likelihood distribution and their standard errors and correlations are written to a parameter output file, while the observed and expected sample statistics and their relative contributions to the likelihood sum are written to a separate output file.

These operations are performed sequentially on all cells in the input data file. Occasionally, some cells return poor fits. To help with finding more accurate descriptions of these cells, the QMPE program allows use in an interactive "trace mode". In this mode, cells may be selected for analysis separately and fit repeatedly. Between each re-fit, almost all parameters of the fit may be altered in the search for a better solution. For example, start points, convergence tolerances, maximum number of search iterations, number and type of sample statistics, and quantile cut-points can all be changed arbitrarily. Coupled with a more detailed search algorithm output, trace mode allows the user much greater control over the fitting process. There is also a "conditional trace mode", which is intermediate between the trace and sequential modes. In this mode, the user can provide a criterion exit code, and sequential fitting of the data file occurs as normal while exit codes are better (i.e., smaller) than the given criterion. When any cell provides a worse fit (i.e., larger exit code) than that specified by the re-fit criterion, QMPE enters trace mode temporarily, allowing the user to "hand-fit" the problematic cell if they desire, before returning to sequential mode and analysing the next cell in the data file.

How to use QMPE

The QMPE program requires two input files: a control file which contains the information QMPE needs to control its behaviour, and a data file which contain the data to be fit. Examples of these files may be found with the source code, on the web site: the example control file is called sample.p and the data file sample.dat.

The Input Files

The data file must be a two-column tab- or space-delimited ASCII file. Each row of this file represents a single data point. The first number on each row is an indexing integer that defines the "cells" of the data file, the second number is the RT sample itself. A "cell" of data is a contiguous block of N samples each of which has the same indexing integer (note that N may change between cells). It is very important that all data points from a cell are contiguous, however it is not necessary for the data points to be in any particular order within a cell.

Any line in the control file which has a hash symbol (#) in the first character position is treated as a comment line: that is, these lines are ignored by the program and are meant to convey information to the user only. In addition to marked comment lines, any line containing a numeric value may have trailing comments as long as they are separated from the number by a space or tab character. The order of arguments supplied in the control file must be adhered to strictly. Disregarding comment lines, the arguments must appear in the following order, without any blank lines in between:

1. The name of the of the input data file. The pathname should be included if the data file is not in the same directory as the program.
2. A stem name for the output files. The suffix ".par" will be appended to this when creating the file that contains the estimated parameters and their standard errors and correlations. The suffix ".oe" will be appended for the file that will contain the observed and expected sample statistics and their contributions to the likelihood sums.
3. A measurement unit for the data file. This is the value of the smallest possible difference between observed data points (e.g., if the data are measured in milliseconds, to a 2ms resolution, enter "2"). This parameter is used to process runs in the data when calculating quantiles.
4. Fitting mode. This takes on either one of three special values, or else a number (greater than 7) indicating the "re-fit exit code criterion. The three special values are: 0 for silent mode in which no output will be written to the standard output except in error states; 1 for normal mode in which one summary line will be printed on the standard output for each cell; 2 for interactive trace mode. If a value greater than 7 is given, QMPE interprets this as a re-fit exit code criterion. In this case, fitting occurs as if mode 1 were in effect (i.e., sequential processing of input file, with one line output per cell). Then, whenever a cell results in an exit code greater than the re-fit criterion, QMPE temporarily enters trace mode, as if mode 2 were in effect. This allows the user to re-fit the problematic cell, and perhaps find a better solution. After re-analysis of that cell is finished, the program continues with analysis of the next cell.
5. Proportional convergence tolerance for the objective function value.
6. Proportional convergence tolerance for the L_∞-norm of the parameter vector. (i.e., proportional convergence tolerance for the largest difference between successive parameter estimates of each parameter.)
7. Maximum number of iterations for the search procedure.
8. Distribution to fit to the data: 1=ExGaussian, 2=Weibull, 3=Shifted LogNormal, 4=Gumbel, 5=Shifted Wald. 6=Gamma. For the Weibull, Shifted LogNormal, Shifted Wald and Gamma distributions, we have implemented the re-parameterisation suggested by Cheng and Iles (1990). We have found that these parameterizations avoid some severe estimation problems associated with nested model (we have a paper under review on this matter). If you want to use the standard parameterizations, use *negative* distribution numbers. The output will be no different in each case.

9. Sample statistics to use, which may take on two values: 1 for when the data are to be used raw (CML); 2 for when quantiles are to be calculated (QMP). If CML fitting is used, no further information is required or read from the control file.
10. Data aggregation level: if 0 the input data will be interpreted as pre-calculated quantiles; if 1 then the maximum possible number of quantiles will be calculated from each cell (N-1 estimates); if 2 then the number following on the next line specifies a fixed number of quantiles to be calculated from each cell; if 3 then the following N_q lines specify arbitrary quantile cut points (between 0 and 1). If data aggregation level is 0 or 1, no further information is required or read from the control file.
11. If data aggregation level (above) was 2, then this line specifies the fixed number of sample statistics to estimate from each cell, this must be smaller than $\min_{cells}\{N-1\}$; no further information is read from the control file. If the data aggregation level was 3, then this line represents the first of the sequence of quantile cut points (up to 1000 points may be specified). N.B. It is very important that the first quantile cut point specified is 0.0 and the final one is 1.0. These values are used as markers by QMPE when reading the parameter file.
12. Any following lines are only read if arbitrary quantile cut points are used (data aggregation level 3). There may be from one to 1000 of these points, but they must be written as a sequence of strictly increasing (not equal) numbers in the interval (0,1), with the value of 0.0 first and 1.0 last.

Program Control

When the program is not to be used in its interactive trace mode (i.e., fitting mode is either 0 or 1), program control is very simple and largely automated. When the program is executed, it first attempts to read the name of the control file from the standard input. There are simple ways to initiate execution and provide the name of the control file in a single step[1], but users can also simply execute the program and then enter the control file name from standard input (e.g., type the name and press enter). The program then reads all required operating parameters from that control file. If this operation executes without error, and the parameters are all valid (e.g. numbers within acceptable ranges, data input files with correct permissions, etc.) the program begins processing the input data. Each data cell is read and processed and then (if not in silent mode) a summary line is printed on the standard output. This line contains a fit number, the size of the data cell (N), the number of sample statistics calculated, then the maximum likelihood estimates of the three parameters μ, σ and τ, followed by estimates of their standard errors and finally an exit code (see Table 1). After each cell is analysed, the results of that analysis are written to the two output files (see below). This process continues until an end-of-file character is reached in the data file.

Exit codes are calculated using a binary addition scheme. For example, and error code of 19 (=1+2+16) means that both convergence tolerances were satisfied at the minimum, and that the best estimate of the third parameter was probably zero[2]. An exit code of 41 (=1+8+32) means that proportional objective function convergence was obtained, that the best estimate of the second parameter was probably zero, that the Hessian matrix evaluated at the maximum likelihood solution was singular, and hence the standard error and correlation estimates are missing for that cell. In our experience, any exit code smaller than 32 means that both the parameter estimates and their standard errors and correlations are trustworthy. Exit codes of 32 and above result in either missing or less

[1] This method was adopted for portability. More elegant solutions, such as passing the name of the control file as a command line argument were tested, but required different versions of the program to be written for different platforms. As QMPE stands now, it can be compiled using any ANSI standard Fortran 90 compiler, on any platform. One-line command execution, as is required in batch file processing, for example, can still be achieved by re-directing the standard input on most operating systems. For example, on both Windows and UNIX, the command “echo sample.p | qmpe” would run the binary “qmpe” and cause it to read the parameter file “sample.p”.

[2] All parameters are constrained to be strictly positive, so that underflows and overflows caused by zero estimates are avoided. See Technical Considerations section for details.

trustworthy standard error and correlation estimates, although the parameter estimates themselves are probably useful as long as the exit code is smaller than 128.

Table 1. Exit Codes.

Code	Meaning	Parameter Estimates	Covariance Estimates
1	Proportional convergence	OK	OK
2	Parameter convergence	OK	OK
4	Best estimate of first parameter is probably zero.	OK	OK if present
8	Best estimate of second parameter is probably zero.	OK	OK if present
16	Best estimate of third parameter is probably zero.	OK	OK if present
32	Hessian is singular.	Probably OK	Missing
64	Maximum number of search iterations reached.	Probably OK	Probably OK
128	Function cannot be evaluated.	Bad	Bad
256	Too few data points to calculate required sample statistics.	Missing	Missing

After analysing an entire data file, the user may wish to try to re-fit some of the cells using different operating parameters and initiating QMPE in its interactive trace mode. This is done by setting the "fitting mode" in the input control file to 2 and initiating the program as above. QMPE will then prompt the user to enter a cell number to analyse. Note that QMPE arbitrarily numbers the cells in the data file in an increasing sequence from 1 for the first cell, 2 for the second, and so forth – without reference to whatever index values were actually used. QMPE then re-reads it's control file (thus implementing any changes the user may have made to that file, except for changes to file names) and load the data from the requested cell. Before fitting the cell, QMPE prompts the user to see whether they wish to use the automatically generated start point or enter one of their own. During fitting, a summary line is output to screen after each iteration of the search algorithm, showing iteration number, parameter values and objective function (negative log-likelihood) value. After the search terminates the user is prompted to either accept the results of that search or re-fit. If the results are accepted, outputs are written to the output files and the user is prompted for the next cell number to analyse. If the results are not accepted, the user may change any of the operating parameters in the control file before the next re-fit.

You can also use a number equal to 8 or greater for the "fitting mode". This will make QMPE run in standard, quiet mode (fitting mode 1) but return to trace mode (fitting mode 2) whenever the exit code from the current cell is equal to or greater than the value specified for fitting mode. This allows the user to re-fit, by hand, only those cells with poor convergence statistics.

The Output Files

Two output files are produced, both are fixed-field-width ASCII files: the output file containing the estimated parameters has suffix ".par"; the file containing observed and expected statistics has suffix ".oe". (Here, and elsewhere, "sample statistic" refers to quantile estimates, if QMP fitting was used, or raw data values, if CML.) The parameter output file has one line per cell, and that line contains 14 numbers. The meanings of these 14 numbers are described in Table 2.

Table 2. Columns in the estimated parameters output file.

Column	Value
1	Cell number: 1 for the first cell in the data file, 2 for the second, and so on.
2	Type of sample statistics used: 1 for raw data, 2 for quantiles.
3	Size of the data cell (N).
4	Number of sample statistics calculated.
5	Exit code.
6	Number of search iterations before termination.
7	Estimated first parameter.
8	Estimated second parameter.
9	Estimated third parameter.
10	Estimated standard error of first parameter.
11	Estimated standard error of second parameter.
12	Estimated standard error of third parameter.
13	Estimated correlation of first and second parameters.

14	Estimated correlation of first and third parameters.
15	Estimated correlation of second and third parameters.
16	Log-Likelihood at best parameter estimates.
17	Cheng & Iles (1990) L0 statistic – useful for detecting nested model problems.
18	Cheng & Iles (1990) L1statistic – useful for detecting nested model problems.
19	Sample mean of the data.
20	Sample standard deviation of the data.

The sample statistics output file (".oe") has five columns, and for each cell in the input data file there are as many lines in this file as sample statistics calculated for that cell. The first column contains simply the cell number. The second column is the quantile cut point used (unless CML fitting was used, in which case this column is filled with zeros). This is a number between zero and one, which was either specified in the input control file (if the data aggregation level was 3) or that was calculated from the number of sample statistics requested. The third column contains the observed sample statistic: the quantile estimate for that cut point. The fourth column contains the corresponding expected sample statistic calculated from the maximum likelihood parameter estimates. The final column gives the contribution of that sample statistic to the log-likelihood sum. The purpose of this file is to allow the detection of outlying data points by either a disproportionately large contribution to the log-likelihood value or else by plotting observed versus expected sample statistics and finding deviations from linearity (i.e., inspection of the QQ plot).

How to Add Other Distributions

The QMPE program has been written in a largely modular format, which makes the task of adapting QMPE to fit any other three parameter[3] distribution relatively easy. There are two different ways to alter the distribution that is fit: a "quick and dirty" and a complete method. The quick and dirty method involves changing only three or four different subroutines and results in almost full functionality but removes the ability of QMPE to fit the ex-Gaussian distribution. The more complete method takes quite a bit more work, but will result in full functionality and will be much more efficient if the CDF of the new distribution has a closed form.

The quick method of fitting a new distribution by CML and QMP methods involves substituting the new distribution's equations for those of the ex-Gaussian at the innermost level. Three subroutines Exgpdf (the density function), DfExgpdf (the derivative of the density function), and GetStartPoint (heuristic parameter estimate) must be altered, and two others are optional (HessExgpdf, the second derivatives of the density, and FitSummaryData, which generates expected values form the fitted distribution function). Changing only the three essential subroutines will result in fully functional CML and QMP fitting, but will not yield applicable Hessian matrices, and will not always give accurate expected statistics in the output file. Changing HessExgpdf will make sure that Hessians are correct and so standard error and correlation estimates will be accurate. Changing FitSummaryData will make sure that the expected sample statistics in the output file (i.e. expected quantile points or raw data values for QQ or PP plots) are correct.

The ease with which different distributions can be accommodated is a result of the fact that the gradients and Hessians of the negative log likelihood functions are simple functions of the gradients and Hessians of the underlying pdf. Appendix A gives the generic expressions which QMPE implements in order to calculate the gradient and Hessian of either the QMP or CML log-likelihood function, based only on the gradient and Hessian values of the underlying pdf. Appendix B gives all expressions used in QMPE that are specific to the ex-Gaussian pdf, and a little information about their derivation. Thus, Appendix B serves as a guide for developing the expressions required to implement QMP fitting for a distribution other than the ex-Gaussian.

Once the above functions are altered, the procedures ObjFunc, DfObjFunc, and HessObjFunc will operate on the outputs of the altered routines to calculate the negative log-likelihood, its gradient and Hessian matrix. With altered routines, maximum numerical efficiency will only be achieved if the new

[3] Throughout QMPE, we have assumed a three-parameter pdf. This allowed greater compuational efficiency, e.g. by assuming that all six independent second-order partial derivatives can be evaluated simultaneously, sharing their many common terms. A two- or one-parameter pdf can easily be implemented by coding with dummy variables.

pdf does not have an analytic form for its CDF (because QMPE will estimate the CDF by numerical integration of the pdf). Improving the efficiency of the function Exgpdf will make the greatest gains in efficiency as it is called far more frequently than any other. The next most frequently called function is DfExgpdf, which is called approximately one order of magnitude less often than Exgpdf.

The most important changes are to the three essential routines. The body of Exgpdf must be replaced so that it returns the value of the pdf for the new distribution given any abscissa value (x) and any parameter vector (par). Similarly, DfExgpdf must be altered such that, given an abscissa value and a parameter vector, it's return argument (df) is a four-element vector: the first element must be the value of the pdf; the next three must be the values of the pdf's partial derivatives with respect to the three elements of the parameter vector. Changing these two routines ensures that the search algorithm operates properly. The function GetStartPoint must be altered so that it returns a parameter vector with which the search algorithm may begin, given any input data. If very little fitting is to be done, this can be accomplished as simply as altering GetStartPoint to request a start point from the user, or use a fixed value. If more fitting is required, heuristics should be developed to automate this process.

Note that because QMPE was developed with the ex-Gaussian distribution in mind, it assumes that the pdf function has support on the entire real line. If the pdf subroutine (Exgpdf) is altered to calculate pdf values for a distribution with bounded support (e.g. the shifted log-normal) it is very important that the routine returns density values of zero for all points outside the support. A small gain in computational efficiency may be achieved by altering the quadrature routines (inside the routines ObjFunc and DfObjFunc) so that numerical integration does not continue outside of the pdf's domain. This gain would probably be small, as the current routines converge very quickly when integrating pdf values that are constantly zero.

In order to get standard error and correlation estimates, the subroutine HessExgpdf must be altered. This subroutine must return a 10-element vector that defines the pdf, it's gradient vector and it's hessian matrix together. The first four elements of this vector are simply the returns from the DfExgpdf routine. The next six are the unique elements of the Hessian matrix[4]. Thus, for a pdf with three parameters, a, b and c, these elements would be: $\frac{\partial^2}{\partial a^2}$, $\frac{\partial^2}{\partial a \partial b}$, $\frac{\partial^2}{\partial a \partial c}$, $\frac{\partial^2}{\partial b^2}$, $\frac{\partial^2}{\partial b \partial c}$, and $\frac{\partial^2}{\partial c^2}$. In order to get accurate expected sample statistics in the output files, some changes must be made to FitSummaryData. This function merely requires that the heuristics which generate the upper and lower bounds for each line search are altered to be correct given the current distribution function. However, as these heuristics have been constructed to be quite robust (e.g. simple step-searching when bounds are unacceptable) there may be no need to alter them.

The complete method for adding a new distribution is more elegant, but requires more coding on the part of the user. For this method, the ex-Gaussian-specific routines are not altered. Rather alterations are made a higher level – to the functions ObjFunc, DfObjFunc, and HessObjFunc. In addition, changes must be made to FitSummaryData and GetStartPoint, as before. In order to keep the functionality of the program, additions are made the pre-existing case structures. Function numbers one to three are reserved for fitting the ex-Gaussian distribution. Thus, for a new pdf, extra cases should be added to ObjFunc, DfObjFunc, and HessObjFunc beginning with case four. These routines should simply return the negative log-likelihood for the data given the new pdf, and the corresponding gradient and Hessian. Finally, extra cases should be added to the controlling subroutine (in the main file QMPE.f90) called DoCell. These changes should set the number of statistics to be extracted and set the exit codes appropriately.

[4] A minor technical assumption here is that the pdf is sufficiently well-behaved that it's mixed second order partial derivatives do not depend on the order of differentiation. This will be true of any pdf with continuously differentiable partial derivatives, for example.

Technical Considerations in QMPE

The Calculation of Quantiles

The source code for QMPE contains a function in the module datagear.f90 that calculates quantiles for a given data sample (called GetQs). In creating this function, we had to make several technical decisions about the calculation of quantiles – estimating quantiles from data is not as unambiguous as may be imagined. The algorithm we have implemented will NOT give the same results as, for example, the S-Plus 2000 function "quantile". The reasons for this lie in the assumptions made about the highest and lowest observed values: the algorithm implemented in S-Plus assumes that the maximum and minimum of the underlying distribution are actually the maximum and minimum sample observations, while our algorithm assumes the distribution to have support on the entire real line. At the code level, this translates into differences in how the quantile cut points (i.e., the numbers between zero and one that define the proportions of data which must fall below each quantile estimate) are mapped onto the indices of the array holding the sorted data. Given N ordered data points in an array indexed by 1,..,N, the S-Plus style algorithm maps a quantile point, q onto an index value by the function: q*(N-1)+1 thus mapping the minimum quantile to the smallest data point, and the maximum quantile to the largest data point. In the same situation, our quantile estimator uses the expression q*N+½, thus mapping the lowest and highest quantile points to below and above the maximum and minimum points. The difference can be seen most strongly in small sample sizes (the two methods converge for large samples). For instance, if three equally spaced quantiles are calculated from a sample {1,2,3,4}, the QMPE algorithm will give 1.5, 2.5 and 3.5, whereas the S-Plus algorithm will give 1.25, 1.5 and 1.75. The first method seems (to us, at least) more sensible when fitting distributions: it seems more likely that 25% of the data falls below 1.5 than below 1.75.

One further technical consideration of the quantile calculator involves data "runs" (i.e. repeated samples of the same value). Runs are problematic for QMP fitting (but not for CML) because a run in the data sample, if sufficiently large, can translate into a run in the estimated quantiles. This in turn means that one of the integrals in Equation 3 will have equal upper and lower limits, so will be zero and hence have an undefined logarithm. To avoid this, the QMPE algorithm pre-processes the data sample by spreading data runs across their limit of measurment accuracy (note that runs can only occur with non-zero probability when the measurement accuracy is finite). For instance, suppose three observations of 500ms were recorded, and the data accuracy was 1ms (as set in the input control file), then those three observations actually fall somewhere between 499.5ms and 500.5ms. To avoid runs, QMPE spreads the three data points evenly across this range, so that {500, 500, 500} becomes {499.75, 500.00, 500.25}.

If the quantile estimator we have implemented is not suited to any particular situation, it can be bypassed by selecting a data aggregation level of zero, and supplying an input data file with quantiles that have be pre-calculated using any algorithm at all. Alternatively, and more permanently, the procedure GetQs can be replaced or augmented as desired.

Parallel Computing

The numerical integrals in Equation 3 can make the use of QMPE very compute-intensive. As such, it is anticipated that this code will sometimes be run on larger shared-memory parallel-processor computers. To aid the use of such computers, compiler directives have been inserted into the code in the objectives.f90 module. Only the single most compute-intensive routine (the call to numerical integration in the objfunc function) has been explicitly parallelised, as this routine accounts for the great majority of total compute time. OpenMP standard compiler directives have been used, as these currently have the widest support: OpenMP directives are currently implemented in compilers on Intel, Sun, SGI, Cray and IBM platforms. Dynamic thread allocation has been explicitly set as the default mode, because the integrals of Equation 3 can be of widely differing computational cost, and fixed thread allocation often provides badly balanced loads in this program. Note that the OpenMP standard will set the total number of threads to eight, or the number of physical processors on the host machine, whichever is smaller. If a different number is required (e.g. more than eight on a large machine, or fewer than the number of processors on a busy machine) then this must be requested at run time by setting the appropriate environment variable (usually either PARALLEL or OMP_NUM_THREADS, but see the Unix manual page "pe_environ(1)" for details).

Technical Notes on the Optimisation Algorithm

The optimisation algorithm uses a conjugate gradient minimiser (adapted loosely from Press et al., 1992). Integrals required for the QMP objective function and it's gradient and Hessian are calculated by adaptive Romberg quadrature. The tolerance for this integration is set as a proportional change tolerance at compile-time, and is currently 10^{-7}. When evaluating integrals for the gradient and Hessian functions, all integrals (four for the gradient, ten for the Hessian) are calculated simultaneously, thus allowing sharing of the many common elements in each and reducing redundancy.

Within the minimiser, all parameters are constrained to be strictly positive by the use of a zero barrier (set at compile time). This barrier is currently set at 10^{-7}. Any parameter values that go below this value result in failed objective function evaluations (i.e., a large negative log likelihood returned, and the setting of a boolean error flag) and gradient evaluations that point back out to the legal region. If the minimser converges to parameter values in an illegal region an exit code notes this. However it is more common that parameter values converge to a value close to the barrier. This situation is also noted by exit codes (values 4, 8 and 16): these are set whenever a parameter estimate falls below 10^{-6}.

Standard error and correlation estimates for the maximum likelihood parameter estimates are calculated from the inverse of the Hessian matrix for the negative log likelihood function (Bates & Watts, 1988; Seber & Wild, 1989). The Hessian is calculated by numerical evaluation of the integrals required. These integrals will occasionally be zero, or near-zero, especially when parameters converge to zero or near-zero estimates, or when the data otherwise under-constrain the parameter estimates (see Bamber & Van Santen, 2000, for a discussion of identifiability). In such cases, the calculated Hessian matrix will be near to singular and so the matrix inversion algorithm (which uses Gauss-Jordan elimination) proceeds in an unstable manner and may produce either numerical exceptions or else unacceptable variance-covariance estimates. In the interests of robustness, tolerances have been set on the sizes of the integrals so that numerical exceptions almost never occur. The price paid for this is that unrealistic covariance estimates occasionally occur, such as correlations outside [-1,1]. QMPE deals with this situation by assuming such values are the result of near-singular Hessian estimates and so are removed from calculation (i.e., all value set to zero) and the exit code for Hessian singularity is set.

References

Bamber, D. & van Santen, J.P.H. (2000) How to assess a model's testability and identifiability. Journal of Mathematical Psychology, 44(1), 20-41.

Bates, D. M., & Watts, D. G. (1988). Nonlinear regression analysis and its applications, New York: Wiley.

Cheng, R.C.H. & Iles, T.C. (1990) Embedded models in three-parameter distributions and their estimation. Journal of the Royal Statistical Society, Series B (Methodological) 52(1) 135-149.

Heathcote, A., Brown, S. & Mewhort, D.J.K. (in press) Quantile Maximum Likelihood Estimation of Response Time Distributions. Psychonomic Bulletin and Review.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. (1992). Numerical recipes in FORTRAN: The art of scientific computing, 2nd Edition, New York: Cambridge University Press

Seber, G.A.F. & Wild, C.J. (1989) Nonlinear regression. New York: John Wiley & Sons.

Van Zandt, T. (2000). How to fit a response time distribution. Psychonomic Bulletin and Review, 7, 424-465.

Appendix A: Gradients and Hessians for Arbitrary PDF

Given a pdf $\underline{f}(t,\theta)$ where $\theta=(\theta_1, \theta_2, \ldots, \theta_p)$ is a parameter vector, and a sample of data $\{x_1,\ldots, x_N\}$, the negative log-likelihood function for CML fitting is given by:

$$-\ln(L) = -\sum_{i=1}^{N} \ln f(x_i, \theta) \qquad \text{(A1)}$$

The gradient of this function is defined by it's partial derivatives with respect to θ_a for a=1,…,p. These can be easily expressed in terms of the corresponding gradient expressions for the pdf $\underline{f}(t,\theta)$:

$$\begin{aligned} -\frac{\partial \ln(L)}{\partial \theta_a} &= -\sum_{i=1}^{N} \frac{\partial}{\partial \theta_a} \ln f(x_i, \theta) \\ &= -\sum_{i=1}^{N} \frac{\frac{\partial}{\partial \theta_a} f(x_i, \theta)}{f(x_i, \theta)} \end{aligned} \qquad \text{(A2)}$$

So to calculate the CML/VML gradient, we require only the gradient of the pdf. The same process may be continued one step further to calculate the Hessian of the objective function. The Hessian is defined by the second partial derivatives, and it can be expressed in terms of the first and second partial derivatives of the pdf:

$$\begin{aligned} -\frac{\partial^2 \ln(L)}{\partial \theta_a \partial \theta_b} &= -\frac{\partial}{\partial \theta_a} \sum_{i=1}^{N} \frac{\frac{\partial}{\partial \theta_b} f(x_i, \theta)}{f(x_i, \theta)} \\ &= -\sum_{i=1}^{N} \frac{\partial}{\partial \theta_a} \frac{\frac{\partial}{\partial \theta_b} f(x_i, \theta)}{f(x_i, \theta)} \\ &= -\sum_{i=1}^{N} \frac{f(x_i, \theta) \cdot \frac{\partial^2}{\partial \theta_a \partial \theta_b} f(x_i, \theta) - \frac{\partial}{\partial \theta_a} f(x_i, \theta) \frac{\partial}{\partial \theta_b} f(x_i, \theta)}{f(x_i, \theta)^2} \end{aligned} \qquad \text{(A3)}$$

Thus, given the gradient and Hessian for the pdf we can calculate the gradient and hessian for the negative log-likelihood, under CML or VML fitting.

If QMP fitting is used, then the quantiles $\{q_1,\ldots, q_{Nq}\}$ are estimated from the data, and the negative log-likelihood function given by:

$$-\ln(L) = -\sum_{i=1}^{N_q+1} n_i \ln\left(\int_{q_{i-1}}^{q_i} f(t, \theta) dt \right) \qquad \text{(A4)}$$

Note that, to simplify notation, we have defined $q_0=-\infty$ and $q_{Nq+1}=+\infty$, and that n_i is the number of observed data points that are inside the interval $[q_{i-1}, q_i]$. Equations corresponding to A2 and A3 above can be derived in almost the same way. One extra result is required: that of Leibnitz' rule which says that:

$$\frac{\partial}{\partial a} \int_c^d f(a,b) db = \int_c^d \left(\frac{\partial}{\partial a} f(a,b) \right) db$$

As long as the limits of integration (c and d) are not functions of the differentiating variable (a). Using this result, it is possible to derive expressions for QMP fitting corrsponding to A2 and A3 which express the gradient and Hessian of the QMP negative log-likelihood function in terms of the gradient and Hessian of the pdf. These are given as Equations A5 and A6:

$$-\frac{\partial \ln(L)}{\partial \theta_a} = -\sum_{i=1}^{N_q+1} n_i \frac{\int_{q_{i-1}}^{q_i} \frac{\partial}{\partial \theta_a} f(t,\theta) dt}{\int_{q_{i-1}}^{q_i} f(t,\theta) dt} \quad \text{(A5)}$$

$$-\frac{\partial^2 \ln(L)}{\partial \theta_a \partial \theta_b} = -\sum_{i=1}^{N_q+1} n_i \frac{\int_{q_{i-1}}^{q_i} f(x_i,\theta) dt \cdot \int_{q_{i-1}}^{q_i} \frac{\partial^2}{\partial \theta_a \partial \theta_b} f(x_i,\theta) dt - \int_{q_{i-1}}^{q_i} \frac{\partial}{\partial \theta_a} f(x_i,\theta) dt \cdot \int_{q_{i-1}}^{q_i} \frac{\partial}{\partial \theta_b} f(x_i,\theta) dt}{\left(\int_{q_{i-1}}^{q_i} f(x_i,\theta) dt\right)^2}$$

(A6)