QUEST+: A general multidimensional Bayesian adaptive psychometric method

Andrew B. Watson

New York University Abu Dhabi, Abu Dhabi,
United Arab Emirates
NASA Ames Research Center, Moffett Field, CA, USA

QUEST+ is a Bayesian adaptive psychometric testing method that allows an arbitrary number of stimulus dimensions, psychometric function parameters, and trial outcomes. It is a generalization and extension of the original QUEST procedure and incorporates many subsequent developments in the area of parametric adaptive testing. With a single procedure, it is possible to implement a wide variety of experimental designs, including conventional threshold measurement; measurement of psychometric function parameters, such as slope and lapse; estimation of the contrast sensitivity function; measurement of increment threshold functions; measurement of noise-masking functions; Thurstone scale estimation using pair comparisons; and categorical ratings on linear and circular stimulus dimensions. QUEST+ provides a general method to accelerate data collection in many areas of cognitive and perceptual science.

Introduction

In this paper, we introduce a general method for efficient adaptive data collection in a broad range of psychometric experiments. The method, which we call QUEST+, is a generalization of the original QUEST procedure. QUEST is an adaptive testing procedure for estimating threshold from a sequence of psychophysical trials (Watson & Pelli, 1979, 1983). It assumes a single stimulus dimension and two possible trial outcomes, and it estimates a single psychometric function parameter that is defined on the stimulus dimension. It also assumes a periodic sampling of the stimulus dimension in the log domain. In comparison, QUEST+

1. Estimates more than one psychometric function parameter
2. Allows more than one stimulus parameter
3. Allows more than two trial outcomes
4. Allows an arbitrary relationship between stimulus and psychometric function parameters
5. Allows an arbitrary sampling of stimulus and parameter dimensions

Since the introduction of QUEST, there has been a number of useful developments in adaptive psychometric methods. Here we briefly review those developments and, when relevant, show how they have been integrated into the QUEST+ procedure. This report does not provide a complete historical review of the topic or a deep mathematical treatment. Some of both are provided by Treutwein (1995).

QUEST is a variant of maximum likelihood (ML) procedures. The history of ML adaptive psychometric procedures begins with Hall (1968, p. 370). In this brief abstract, he laid out the fundamental idea:

> The experimenter assumes a parametric form of the psychometric function … After each response, values of the parameters are computed that maximize the probability of the set of responses that have been obtained, given the set of stimuli that have been presented. These values are then used to determine the placement of the next stimulus.

The innovations of the QUEST procedure were (a) to introduce the idea of a prior probability density for the psychometric function parameters, thus rendering the procedure Bayesian, and (b) to show that the Bayesian posterior density for threshold could be computed by merely shifting and adding lists of precomputed log

Citation: Watson, A. B. (2017). QUEST+: A general multidimensional Bayesian adaptive psychometric method. *Journal of Vision*, *17*(3):10, 1–27, doi:10.1167/17.3.10.

doi: 10.1167/17.3.10 Received April 19, 2016; published March 29, 2017 ISSN 1534-7362 Copyright 2017 The Authors

Downloaded from jov.arvojournals.org on 11/02/2020

This work is licensed under a Creative Commons Attribution 4.0 International License.

likelihoods for success and failure outcomes. This simplification relied on an assumption that the psychometric function did not change in slope as the threshold parameter varied.

The original version of QUEST placed the next trial at the mode of the posterior density. Emerson (1986) and King-Smith, Grigsby, Vingrys, Benes, and Supowit (1994) subsequently showed through simulation that the mean of the density was a better choice.

The original QUEST estimated only the threshold parameter of the psychometric function although Watson and Pelli (1983) did note that the Bayesian method could be extended to estimate additional parameters. King-Smith and Rose (1997) and Snoeren and Puts (1997) implemented this extension of the Bayesian approach to estimation of the slope of the psychometric function as well as threshold. Generalization of parameter estimation to two dimensions was straightforward, but both reports acknowledged that the more difficult problem was placement of the next trial. Snoeren and Puts extended the concept of the "sweat factor" to two dimensions for a certain class of psychometric functions, which led to particular rules for trial placement. King-Smith and Rose used heuristic placement rules to emphasize threshold or slope estimation.

An alternative idea for trial placement is to minimize the expected variance of the parameter estimate (King-Smith et al., 1994). Extending this to two dimensions is problematic, however, because the variance of each parameter will depend on the units of its representation. Instead, following a suggestion of Pelli (1987), Kontsevich and Tyler (1999) adopted a placement rule that minimized the expected entropy of the two-dimensional posterior probability density of the parameter estimates. Unlike variance, entropy is not altered by a change of scale. Pelli had proposed looking ahead through all possible future trial sequences, but King-Smith et al. (1994) showed that for variance minimization a single step look ahead was sufficient, and that was the rule adopted by Kontsevich and Tyler.

In addition to generalizing the Bayesian approach to more than one psychometric function parameter, it is also possible to generalize to more than one stimulus parameter. The traditional color matching experiment, in which an observer adjusts the intensities of a set of three primaries that are combined to match the color of a test light, is an example. Two additional examples are the rapid estimation of the contrast sensitivity function (Lesmes, Lu, Baek, & Albright, 2010), in which the stimulus dimensions are contrast and spatial frequency, and threshold versus noise masking, in which the stimulus dimensions are contrast and noise energy (Lesmes, Jeon, Lu, & Dosher, 2006).

These sorts of data can be accommodated within the Bayesian framework. It is only a matter of expanding the dimensions of the priors, likelihoods, and posteriors in appropriate ways. And the minimum entropy rule (Kontsevich & Tyler, 1999) again provides a guide to trial placement in the now multidimensional stimulus space.

Expanding the number of stimulus dimensions has also been explored by Kujala and Lukka (2006), Vul, Bergsma, and MacLeod (2010); and DiMattina (2015). These efforts have also used ML or Bayesian estimation and entropy minimization for trial placement but have pursued complex or less general techniques to accelerate the procedure. We have not pursued those methods here, but we discuss computation time for our general method below.

In QUEST+, we also consider a further generalization of typical adaptive procedures. The QUEST procedure was designed to estimate a single parameter of a hypothetical psychometric function and to do so by selecting testing points along a single stimulus dimension. In the most common example, the function parameter was threshold, and the stimulus parameter was stimulus intensity. These two quantities were assumed to be values on the same dimension. But that is an arbitrary constraint. Within the Bayesian framework, it is possible to have psychometric function parameters that are distinct from the stimulus parameters.

The final generalization of QUEST+ is to allow more than two trial outcomes. In QUEST and the other methods discussed above, these two outcomes are *failure* and *success*, *incorrect* and *correct*, *no* and *yes*, or another congruent pair. But the Bayesian framework is not limited to that case. Below we show how other psychometric procedures, such as categorization, with more than two possible outcomes can be handled by QUEST+.

Theory

Here we provide a brief mathematical derivation of the QUEST+ procedure. This is a generalization of previous accounts (Kontsevich & Tyler, 1999; Watson & Pelli, 1983), so we provide only a brief description emphasizing new extensions and generalizations to additional stimulus and outcome dimensions.

The vector of stimulus parameters is written $\mathbf{x}$, the vector of psychometric function parameters is written $\mathbf{s}$, and the outcome is written r. The psychometric function itself is written $\Psi(\mathbf{x},\mathbf{s})$. In contrast to previous treatments, here

Downloaded from jov.arvojournals.org on 11/02/2020

the psychometric function always returns a vector of probabilities, corresponding to the several possible outcomes r. For a given $\mathbf{x}$ and $\mathbf{s}$, those probabilities must sum to one. Alternatively, we can express the psychometric function as the conditional probability of outcome r given $\mathbf{x}$ and $\mathbf{s}$ or $p(r|\mathbf{x}, \mathbf{s})$. The stimulus value on trial k is a vector $\mathbf{x}_k$, and the complete sequence of stimulus values is $\mathbf{X}_k = \{\mathbf{x}_1, \mathbf{x}_2,\ldots, \mathbf{x}_k\}$. The corresponding sequence of responses is $\mathbf{r}_k = \{r_1, r_2, \ldots, r_k\}$. Although it is arbitrary, we generally consider the several possible outcomes to be identified by successive integers starting at one (e.g., $r = 1$ -> failure, $r = 2$ -> success). We adopt the convention of using uppercase P to indicate probability density functions and lowercase p to indicate likelihoods or conditional probabilities. We also use the prime symbol to indicate un-normalized density functions; the corresponding symbol without a prime is understood to have been normalized by dividing by its integral.

We assume an observer who is statistically stationary and for whom successive trials are independent. The prior probability density function (PDF) of the psychometric parameters is written $P(\mathbf{s})$. From Bayes' theorem, we can write the posterior probability of the parameters after trial k as the product of the prior density and the likelihood of the data

$$P'_k(\mathbf{s}) = P'(\mathbf{s}|\mathbf{X}_k, \mathbf{r}_k) = P(\mathbf{s})p(\mathbf{r}_k|\mathbf{X}_k, \mathbf{s}) \quad (1)$$

$$p(\mathbf{r}_k|\mathbf{X}_k,\, \mathbf{s}) = \prod_k p(r_k|\mathbf{x}_k, \mathbf{s}) \quad (2)$$

Thus, the posterior density after trial k + 1 is given by

$$P'_{k-1}(\mathbf{s}) = P(\mathbf{s})p(\mathbf{r}_k|\mathbf{X}_k, \mathbf{s})p(r_{k+1}|\mathbf{x}_{k+1}, \mathbf{s}) = P'_k(\mathbf{s})p(r_{k+1}|\mathbf{x}_{k+1}, \mathbf{s}) \quad (3)$$

In words, we multiply the posterior density on trial k by the likelihood of trial $k + 1$.
After trial k, we can compute, for any value of $\mathbf{x}$, the probability of outcome r

$$p_k(\mathbf{r}|\mathbf{x}) = \sum_k P_k(\mathbf{s})p(r|\mathbf{x}, \mathbf{s}) \quad (4)$$

(We use a sum rather than an integral to foreshadow our subsequent discrete treatment of the problem.) The entropy of a particular posterior density is given by

$$H = \sum_k P(\mathbf{s}) \log[P(\mathbf{s})] \quad (5)$$

Thus, after trial k, we can write a function that describes the entropy that would result from outcome r at stimulus $\mathbf{x}$

$$H_k(r, \mathbf{x}) = \sum_k P_k(\mathbf{s})p(r|\mathbf{x}, \mathbf{s}) \log[P_k(\mathbf{s})p(r|\mathbf{x}, \mathbf{s})] \quad (6)$$

Finally, the expected entropy of a trial at x is given by

$$EH_k(\mathbf{x}) = \sum_j p_k\left(r_j|\mathbf{x}\right)H_k\left(r_j, \mathbf{x}\right) \quad (7)$$

The minimum entropy procedure consists of selecting the value of $\mathbf{x}$ that minimizes the expected entropy EH.

Implementation

Mathematica

We have implemented the QUEST+ procedure in the Mathematica language. Here we provide a brief description of the structure and syntax. The code is available for download. In Mathematica, function arguments are enclosed in square brackets; vectors are represented by lists, which are enclosed in curly braces; and arrays are represented by lists of lists.

Downloaded from jov.arvojournals.org on 11/02/2020

Configuration

The QUEST+ method is highly general and can be applied to a wide variety of experiments through simple configuration. There are six steps in the configuration of the QUEST+ method:

1. Specification of the stimulus parameter domain
2. Specification of the psychometric function
3. Specification of the psychometric function parameter domain
4. Specification of the outcome domain
5. Definition of the outcome function
6. Specification of the prior probability of parameter values

Psychometric function

First, we describe the syntax of psychometric functions. They have a name, a list of stimulus parameters, and a list of psychometric parameters. They always return a list corresponding to the probabilities of the several possible outcomes. This list has a length ≥ 2 equal to the number of possible outcomes. We use lists for stimulus parameters, psychometric parameters, and outcomes because this makes it easy to treat multidimensional parameters, stimuli, and outcomes. Here is the example of a Weibull psychometric function that includes one stimulus parameter, four psychometric parameters, and two outcomes. In this example, **c** and **threshold** are contrasts in dB, and **slope**, **guess**, and **lapse** are the traditional parameters of the Weibull function (Treutwein, 1995; Watson & Pelli, 1983).

```
QpPFWeibull[{c_}, {threshold_, slope_, guess_, lapse_}] :=
{#, 1 - #} &@(lapse - (guess + lapse - 1) Exp[-10^(slope (c - threshold)/20) ])
```

When we configure the experiment, we use the Mathematica option construct to tell QUEST+ which psychometric function to use. In this case, the option would be

```
QpPsi -> QpPFWeibull
```

Outcome function

In actual use, the experimenter creates a function that presents a stimulus to the observer, collects a response, and returns an outcome. The general syntax is **Outcome[x_List]**, where **x** is a list of stimulus parameters and the function must return one of the possible outcomes. The outcomes are identified by positive integers starting at one. The number of possible outcomes *n* is determined by the psychometric function. For the Weibull function above, $n = 2$.

To demonstrate the QUEST+ procedure, it is useful to have a simulated observer. We provide a function that creates a simulated observer obeying a specified psychometric function.

```
SimulatedObserver[x_List, psi_, s_List] :=
 Position[RandomVariate[MultinomialDistribution[1, psi[x, s]]], 1][[1, 1]]
```

The three arguments are a list of stimulus parameters **x**, the name of a psychometric function **psi**, and a list of psychometric parameter definitions **s**. This function makes use of the multinomial distribution, which is a generalization of the binomial distribution. The psychometric function returns the *n* probabilities of the *n* possible responses, and the SimulatedObserver function randomly returns a number between one and *n* with the specified probabilities.

In this simulation, we use a simulated observer following a Weibull psychometric function with parameters fixed at specific values. To configure the experiment, we tell QUEST+ which outcome function to use with an option

```
QpOutcome -> (SimulatedObserver[#, QpPFWeibull, {-20, 3, .5, .02}] & )
```

In Mathematica, the "&" symbol terminates a temporary function with an argument given by #.

Downloaded from jov.arvojournals.org on 11/02/2020

Stimulus parameters

In a discrete implementation, we must specify the permissible sample values of the stimulus parameters **x**. In our implementation, we do this by specifying the actual samples in each dimension. We use the Mathematica option construct to specify these quantities. Here is an example in which we indicate that the single stimulus parameter should take values from –40 to zero in steps of one. Note that the structure is a list of lists because QUEST+ allows more than one stimulus parameter.

```
QpXSamples -> {Range[-40, 0]}
```

Psychometric parameters

In a similar way, we specify sample lists for the psychometric parameters **s**. In this case, we vary two parameters: threshold and slope. But the Weibull psychometric function defined above has four parameters. Here we handle that by specifying single values (lists of length one) for the fixed parameters. So the specification is a list of four lists. The first is for threshold and the second for slope, and the remaining two are fixed values for guess and lapse.

```
QpPSamples -> {Range[-40, 0], Range[2, 5], {.5}, {0.02}}
```

It is also possible to achieve the same result with a new definition of the Weibull in which guess and lapse are fixed, but our current approach has the advantage of simplicity and generality at no additional cost.

Priors

The specification of priors is accomplished through the **QpPriors** option. The default setting is

```
QpPrior -> “Constant”
```

which indicates that the prior should have a uniform probability over all combinations of the parameter values. Alternatively, the user can provide an explicit prior over the domain of the psychometric parameters.

Configuration options

We configure an experiment by setting all of the options described above, specifying the stimulus and parameter samples, the outcome function, the psychometric function to be fit to the data, and the prior. The following example simply groups together the options discussed above for the case of estimation of two psychometric parameters: threshold and slope.

```
options = Sequence[
  QpOutcome -> (SimulatedObserver[#, QpPFWeibull, {-20, 3, .5, .02}] & )
 , QpPNames -> {"Threshold", "Slope", "Guess", "Lapse"}
 , QpXNames -> {"Contrast (dB)"}
 , QpXSamples -> {Range[-40, 0]}
 , QpPSamples -> {Range[-40, 0], Range[2, 5], {.5}, {0.02}}
 , QpPsi -> QpPFWeibull
 , QpPrior -> "Constant"];
```

Some of these options need not be specified, such as **QpPNames** and **QpXNames**, as they are used only for subsequent analysis and plotting or are defaults, such as **QpPrior**, but we show them here for completeness.

Downloaded from jov.arvojournals.org on 11/02/2020

QuestPlus

The primary function is QuestPlus. It is called with a single argument—the number of trials—and possibly a sequence of configuration options. Here is an example using the configuration defined above.

```
result = QuestPlus[32, options]
```

```
{{-20,3,0.5,0.02},{{{-18},2},{{-22},1},{{-12},2},{{-13},2},{{-15},2},{{-16},2},{{-17},2},{{-18},2},{{-19},2},{{-
   21},2},{{-24},2},{{-25},2},{{-26},2},{{-27},1},{{-21},2},{{-22},2},{{-23},1},{{-20},2},{{-20},2},{{-20},2},{{-
   21},1},{{-19},2},{{-19},2},{{-20},2},{{-20},2},{{-20},1},{{-19},2},{{-19},2},{{-19},2},{{-19},2},{{-19},2},{{-
   20},2},{{-20},2},{{-20},2},{{-20},2},{{-20},2},{{-21},2},{{-21},1},{{-20},2},{{-20},2},{{-20},1},{{-19},1},{{-
   18},2},{{-18},2},{{-19},2},{{-19},2},{{-19},1},{{-18},2},{{-18},1},{{-17},2},{{-18},2},{{-18},2},{{-18},1},{{-
   17},2},{{-17},2},{{-17},2},{{-17},2},{{-17},2},{{-17},2},{{-17},2},{{-18},2},{{-18},2},{{-18},2},{{-18},2}}}
```

The structure of the output is a list whose first element is the list of estimated and fixed psychometric function parameters. This estimate is determined by the maximum (mode) of the posterior density and is, of course, quantized to the resolution defined by QpPSamples. The remainder is the sequence of trials, each expressed as a list {stimulus parameters, outcome}. We can plot these trials as percentage correct along with the estimated psychometric function as shown in Figure 1. We use color saturation of the data points to reflect the number of trials at each intensity. This plot also shows the quantized parameter estimates as well as a more precise estimate obtained from a subsequent ML fit.

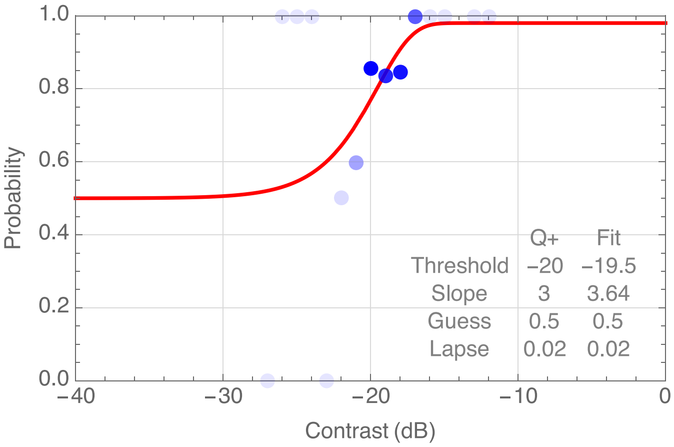


Figure 1. Result of a simulated experiment with one stimulus parameter (intensity) and two psychometric parameters (threshold and slope). The color saturation of the data points corresponds to the number of trials. The parameter estimates from QuestPlus and from subsequent ML fitting are shown.

Alternate version

The **QuestPlus** function above assumes that the user provides only a single additional function: the function that presents the stimulus to the observer. In that sense, **QuestPlus** is the "master" program. Users may prefer to write their own master program and invoke QUEST+ functions as subordinate routines. This is useful when a number of conditions are to be interleaved, when stimulus creation and presentation are complex, or when the user would like to deviate from the stimulus parameters for the next trial recommended by QUEST+. We provide this capability in the form of four functions: **QpInitialize**, **QpQuery**, **QpUpdate**, and **QpRun**. The last is only an example of how to invoke the first three, and we show it here in its entirety.

Downloaded from jov.arvojournals.org on 11/02/2020

```
QpRun[ntrials_, options___Rule] :=Block[{outcomefunction, outcome, structure, psamples},
  {outcomefunction, psamples} = {QpOutcome, QpPSamples} /. {options} /. Options[QuestPlus];
  structure = QpInitialize[options];
  Do[(
    structure=QpQuery[structure];
    outcome=outcomefunction[structure[[5]]];
    structure=QpUpdate[structure, outcome];)
  ,{ntrials}];
  {MapThread[#1[[#2]]&,{psamples, ListArgMax[structure[[2]]]}],structure[[1]]}]
```

Like **QuestPlus**, **QpRun** returns the quantized parameter estimates and the data. **QpInitialize** creates a data structure that is subsequently queried and updated on each trial. When several conditions are interleaved, **QpInitialize** would be called once for each to create an array of structures. **QpQuery** determines the recommended next stimulus vector, and **QpUpdate** uses the previous outcome to update the data structure.

Fitting and plotting the results

Although not strictly part of the QUEST+ procedure, we note here several functions for fitting and plotting the results. The function **QpFit** fits a psychometric function to the result. This will typically be the psychometric function used in the QUEST+ procedure and specified in the options. Although QUEST+ effectively finds the ML fit on every trial, **QpFit** is not constrained to the quantized parameter space. But because it takes advantage of the same estimation logic, it operates on any number of stimulus dimensions, psychometric parameters, and outcomes.

We also provide plotting functions **QpPlot**, **QpPlot2D**, **QpPlot3D**, and **QpPlotCategories** to show the results of one, two, and three stimulus dimensions and of categorization (rating) experiments. These functions first fit the data using **QpFit** and then plot data and best-fitting psychometric function. Examples of these plots are shown below.

Examples

Here we provide some examples of the use of QUEST+. After the title of each example, we indicate as a list of numbers the following: the number of stimulus dimensions, the number of parameter dimensions, and the number of responses. These examples can also be run live using the demonstration described in the appendix.

Estimation of contrast threshold {1, 1, 2}

The first is a very simple case in which there is a single stimulus dimension (contrast), and only a single psychometric parameter (threshold). This is the case for which the original QUEST was designed. This is the default configuration for QUEST+, so we examine the default options:

```
Options[QuestPlus] =
  { QpOutcome -> (SimulatedObserver[#, QpPFWeibull, {-20, 3.5, .5, .02}] &)
  , QpPNames -> {"Threshold (dB)", "Slope", "Guess", "Lapse"}
  , QpXNames -> {"Contrast (dB)"}
  , QpXSamples -> {Range[-40, 0]}
  , QpPSamples -> {Range[-40, 0], {3.5}, {.5}, {0.02}}
  , QpPsi -> QpPFWeibull
  , QpPrior -> "Constant"
  , Verbose -> False};
```

Then we run QuestPlus. Because we are using defaults, we do not need to specify any options.

Downloaded from jov.arvojournals.org on 11/02/2020

```
result = QuestPlus[32]
```

```
{{-20, 3.5, 0.5, 0.02}, {{{-18}, 2}, {{-22}, 2}, {{-25}, 2}, {{-28}, 2}, {{-30}, 1}, {{-22}, 1}, {{-13}, 2}, {{-15}, 2},
  {{-16}, 2}, {{-18}, 2}, {{-19}, 2}, {{-20}, 2}, {{-21}, 2}, {{-22}, 2}, {{-23}, 1}, {{-19}, 2}, {{-20}, 2}, {{-20}, 1},
  {{-18}, 2}, {{-18}, 2}, {{-19}, 1}, {{-17}, 2}, {{-17}, 2}, {{-18}, 2}, {{-18}, 2}, {{-18}, 2}, {{-19}, 2}, {{-19}, 2},
  {{-19}, 2}, {{-19}, 2}, {{-19}, 2}, {{-19}, 2}}}
```

Recall that the first element in the output is the list of quantized estimated parameters. Note that they are equal to the simulated values. We plot the trials as proportion correct along with the estimated psychometric function, as shown in Figure 2, produced by the following command.

```
QpPlot[result]
```

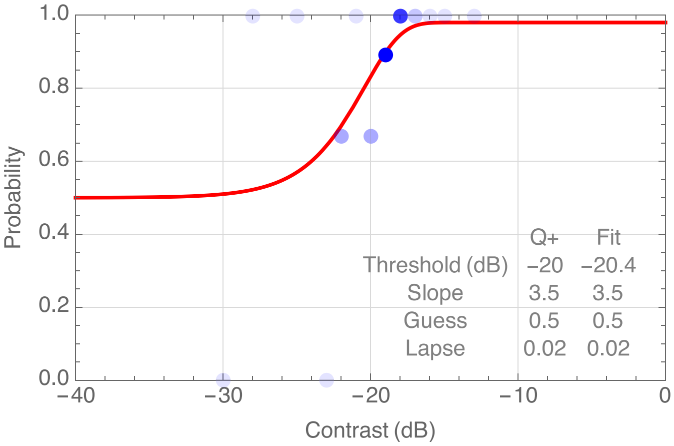


Figure 2. Result of a simulated experiment with one stimulus parameter and one psychometric parameter. The color saturation of the data points corresponds to the number of trials. The parameter estimates from QUEST+ and from subsequent ML fitting are shown.

Estimation of contrast threshold, slope, and lapse {1, 3, 2}

In this case, we again have only a single stimulus parameter (contrast) but three psychometric parameters (threshold, slope, and lapse). We define suitable options. For the simulated observer, we fix the parameter values of the Weibull psychometric function. For the model function, we use a version of the Weibull in which only guess is fixed at 0.5. We choose suitable samples for the three parameters.

```
options = Sequence[
  QpOutcome -> (SimulatedObserver[#, QpPFWeibull, {-20, 3, .5, .02}] & )
  , QpXNames -> {"Contrast (dB)"}
  , QpPNames -> {"Threshold (dB)", "Slope", "Guess", "Lapse"}
  , QpXSamples -> {Range[-40, 0]}
  , QpPSamples -> {Range[-40, 0], Range[2, 5], {.5}, Range[0, .04, .01]}
  , QpPsi -> QpPFWeibull];
```

Because we are estimating three parameters, we run 64 trials,

```
result = QuestPlus[64, options];
```

and we plot the result in Figure 3. The estimated parameters are close to the simulated values.

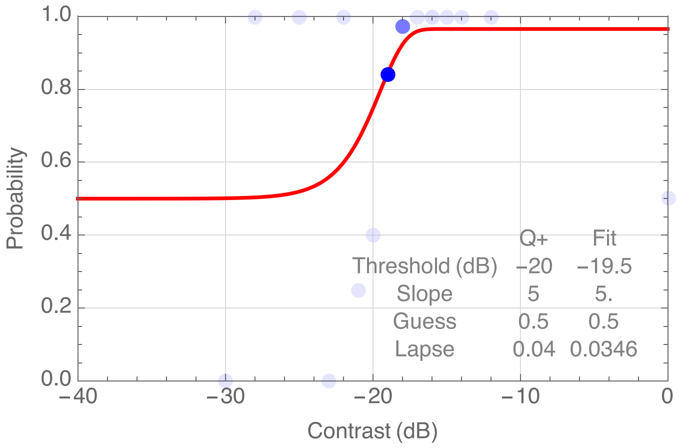


Figure 3. Result of a simulated experiment with one stimulus parameter and three psychometric parameters. The color saturation of the data points corresponds to the number of trials. The parameter estimates from QUEST+ and from subsequent ML fitting are shown.

Estimation of mean, standard deviation, and lapse {1, 3, 2}

This case arises often in the context of a comparison between a standard and a test stimulus. For example, the standard might be a line at some orientation and the test a line subsequently displayed at a possibly different orientation. The observer's task is to say whether the test is rotated clockwise or counterclockwise to the standard (Skottun, Bradley, Sclar, Ohzawa, & Freeman, 1987). We define a response of "clockwise" as "success." If the stimulus dimension is rotation clockwise relative to the standard, then the psychometric function will increase for positive rotations and decrease for negative. This psychometric function could be modeled as a cumulative Normal distribution. The mean would reflect a bias, and the standard deviation would reflect sensitivity. An important feature of this case is that we fully expect the slope to change with conditions because it is a measure of sensitivity. Thus, a possibly large parameter range must be allowed for standard deviation with possibly smaller ranges for mean and lapse.

We write the psychometric function for this case as

```
QpPFNormal[{x_},{mean_, sd_, lapse_}] := lapse+(1-2 lapse) CDF[NormalDistribution[mean, sd], x]
```

and set options as

```
options = Sequence[
  QpOutcome -> (SimulatedObserver[#, QpPFNormal, {1, 3, .02}] & )
 , QpXNames -> {"Orientation (°)"}
 , QpPNames -> {"Mean (°)", "SD (°)", "Lapse"}
 , QpXSamples -> {Range[-10, 10]}
 , QpPSamples -> {Range[-5, 5], Range[1, 10], Range[0, .04, .01]}
 , QpPsi -> QpPFNormal];
```

We again run 128 trials,

```
result = QuestPlus[128, options];
```

and plot the result in Figure 4. Note that to estimate the standard deviation, many trials are placed above and below the mean.

Downloaded from jov.arvojournals.org on 11/02/2020

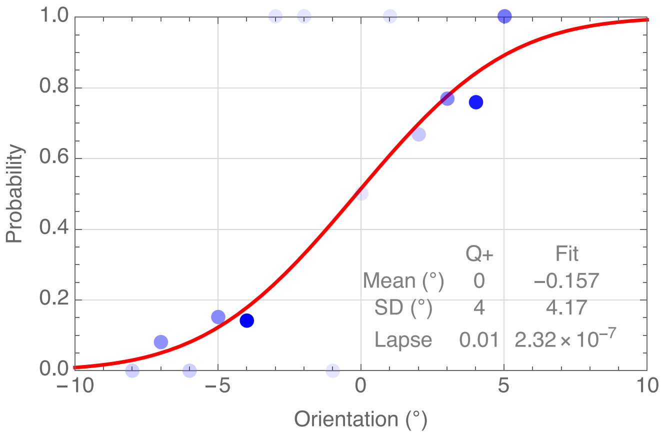


Figure 4. Plot of psychometric data from estimation of mean, standard deviation, and lapse in a comparison experiment.

Contrast sensitivity function {2, 3, 2}

In this example, we illustrate a case of multiple stimulus dimensions. The example describes rapid estimation of a parametrically defined contrast sensitivity function. Our approach is effectively the same as that of Lesmes et al. (2010) except that we use a different model for the contrast sensitivity function. Our main point is to show that their qCSF method can be easily accommodated into the QUEST+ framework.

We have recently reported that, at moderate photopic luminances and medium to high spatial frequencies, log contrast sensitivity is a linear function of spatial frequency (Watson & Ahumada, 2016). At low frequencies, sensitivity stays constant or declines. This model can be written

$$c = \max\left[t, c_0 + c_f f + c_w w\right] \quad (8)$$

where c is the contrast threshold in dB, f is spatial frequency in cycles per degree, w is temporal frequency in Hz, t is the minimum threshold, and c_0, c_f, and c_w are the linear coefficients. If we set $w = 0$, we obtain the spatial contrast threshold function (the inverse of the contrast sensitivity function) as shown in Figure 5A.

A psychometric function based on this threshold can be specified as follows. This makes use of our previous definition of a Weibull psychometric function, and we fix its slope, guess, and lapse parameters at 3, 0.5, and 0.01.

```
QpPFSCSF[{f_, c_}, {t_, c0_, cf_}] := QpPFWeibull[{c}, {Max[t, c0 + cf f], 3., .5, .01}]
```

In words, this is a Weibull psychometric function with constant slope of three, constant guess of 0.5, and constant lapse of 0.01 and a threshold in dB given by Equation 8. This function is portrayed as an image in Figure 5B.

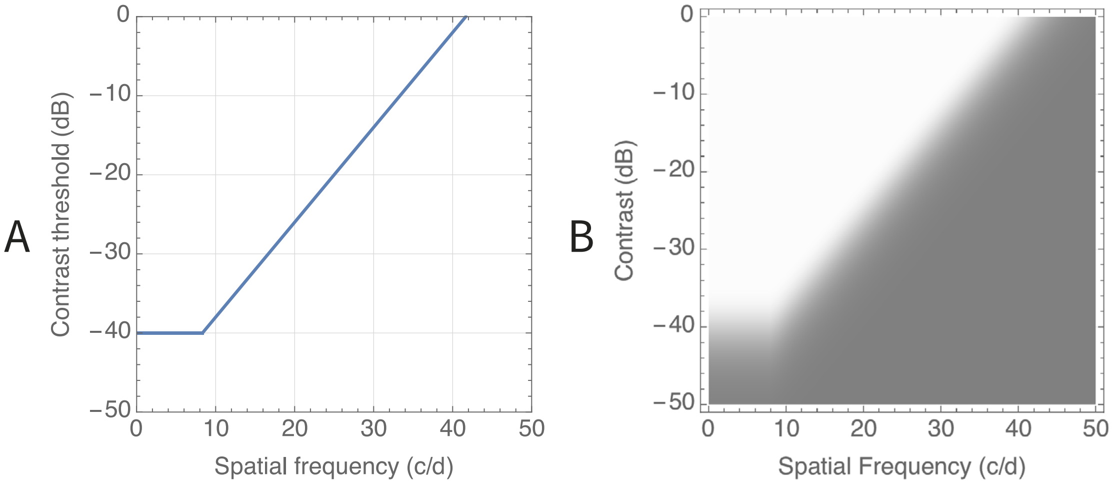


Figure 5. (A) A linear model for the contrast threshold as a function of spatial frequency. (B) Two-dimensional psychometric function based on the linear model of the spatial contrast sensitivity function.

Downloaded from jov.arvojournals.org on 11/02/2020

This condition for QUEST+ can be specified by options that determine the simulated observer and its psychometric function; the two lists of samples for stimulus contrast and spatial frequency; the three lists of samples for the parameters c_0, c_f, and t; and finally, the psychometric function whose parameters are to be estimated.

```
options = Sequence[
   QpOutcome -> (SimulatedObserver[#, QpPFSCSF, {-35, -50, 1.2}] & )
  , QpPNames -> {"t", "c0", "cf"}
  , QpXNames -> {"Spatial frequency (c/d)", "Contrast (dB)"}
  , QpXSamples -> {Range[0, 40, 2], Range[-50, 0, 2]}
  , QpPSamples -> {Range[-30, -50, -2], Range[-40, -60, -2], Range[.8, 1.6, .2]}
  , QpPsi -> QpPFSCSF];
```

We run a simulation of 32 trials as follows.

```
result = QuestPlus[32, options]
```

```
{{-32,-56,1.4},{{{40,0},2},{{40,-4},1},{{34,0},2},{{36,0},2},{{0,-38},2},{{38,0},2},{{0,-40},1},{{38,0},2},{{18,-
   26},2},{{18,-26},2},{{40,0},2},{{0,-36},2},{{0,-36},1},{{40,0},2},{{20,-26},2},{{20,-26},2},{{40,-2},2},{{22,-
   26},2},{{40,-6},1},{{20,-26},1},{{40,0},2},{{18,-26},2},{{40,0},2},{{18,-26},2},{{0,-32},2},{{0,-32},2},{{0,-
   34},2},{{0,-34},1},{{40,0},1},{{18,-26},2},{{38,0},2},{{18,-26},2}}}
```

The output is a list. The first element is a list of estimated parameters, and the second is the sequence of trials, each expressed as {{f,c},r}, where r is one or two. Note that with only 32 trials the estimated parameters are close to the simulated values.

We can plot this sequence on top of the psychometric function of the simulated observer to see where QUEST+ placed its trials as shown in Figure 6. We use disc symbols in which size indicates number of trials, and the angular portion in green represents the percentage correct. It is clear that the trials are placed effectively to estimate the three parameters at the three critical points on the function, and few trials are wasted in uninformative locations.

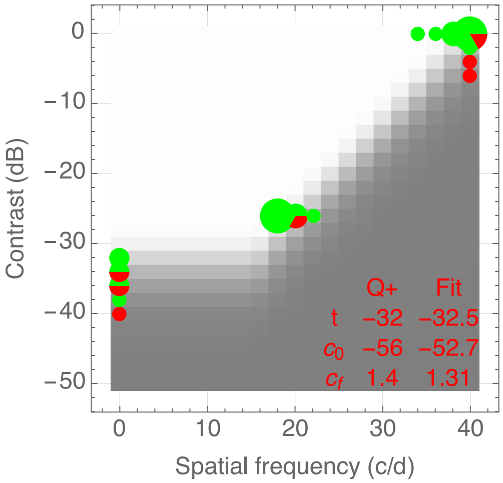


Figure 6. Placement of QUEST+ trials. The simulated two-dimensional psychometric function is also shown along with quantized and fit parameters.

We can also examine the final posterior density. Because it is three-dimensional, we show in Figure 7 only the slice at the peak value of the parameter c_f.

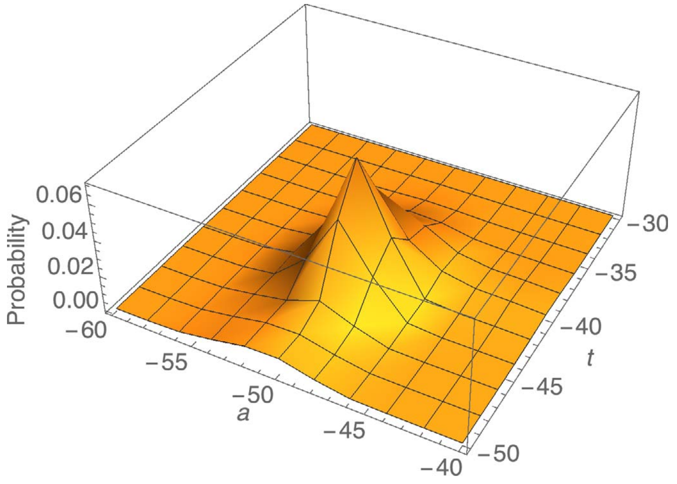


Figure 7. The posterior density for the parameters c_0 and t at the value $c_f = 1.2$, from a simulated estimation of the CSF.

Spatiotemporal contrast sensitivity {3, 4, 2}

We can extend the previous example to three stimulus dimensions. Referring again to Equation 8, we note that log contrast sensitivity is a linear function of both spatial and temporal frequency (Watson & Ahumada, 2016). An example is shown in Figure 8.

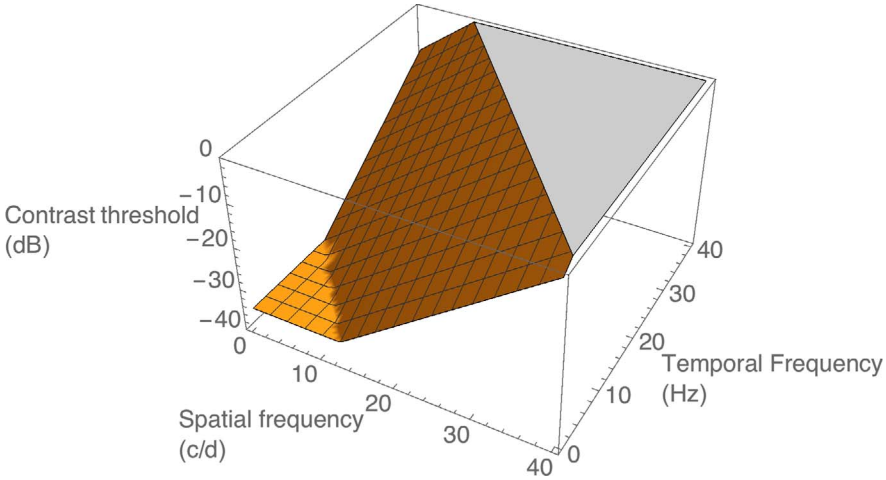


Figure 8. Spatiotemporal contrast sensitivity function based on the Pyramid of Visibility (Watson & Ahumada, 2016). The parameters are $t = -35$, $c_0 = -50$, $c_f = 1.2$, $c_w = 1$.

This linear model allows us to construct a psychometric function that is a function of three stimulus parameters (contrast, spatial frequency, and temporal frequency) and that has three psychometric parameters.

```
QpPFSTCSF[{c_, f_, w_}, {t_, c0_, cf_, cw_}] := QpPFWeibull[{c}, {Max[t, c0 + cf f + cw w], 3., .5, .01}]
```

We can now configure an adaptive experiment with the following options:

```
options=Sequence[
  QpOutcome-> (SimulatedObserver[#, QpPFSTCSF,{-40,-50,1.2,1.0}]& )
, QpPNames->{"t","c0","cf","cw"}
, QpXNames->{"Contrast (dB)","Spatial frequency (c/d)","Temporal frequency (Hz)"}
, QpXSamples->{Range[-40,0,5], Range[0,40,5], Range[0,40,5]}
, QpPSamples->{Range[-30,-50,-5], Range[-40,-60,-5], Range[.8,1.6,.2], Range[.8,1.6,.2]}
, QpPsi-> QpPFSTCSF];
```

And we can now execute 64 simulated trials.

```
(result = QuestPlus[64, options])//Short
```

```
{{−35,−50,1.2,1.},{{<1>,2},<<62>>,{<1>}}}
```

We note that the quantized estimates of the four parameters are reasonable. We plot the results, using the function **QpPlot3D**. In order to see the placement of the trials, we show the three-dimensional stimulus space as a series of plots, one for each of the contrast samples. Each location tested is plotted as a disc with a size reflecting the number of trials at that location, and the angular portion in green represents the percentage correct. We superimpose these points on gray-level representations of the psychometric function of the simulated observer. This representation of the results is shown in Figure 9. QUEST+ again concentrates the trials at the critical locations in the three-dimensional stimulus space. In 64 trials, of the 729 possible stimulus locations, only 20 are visited. In a subsequent simulation of 256 trials, only 24 were visited.

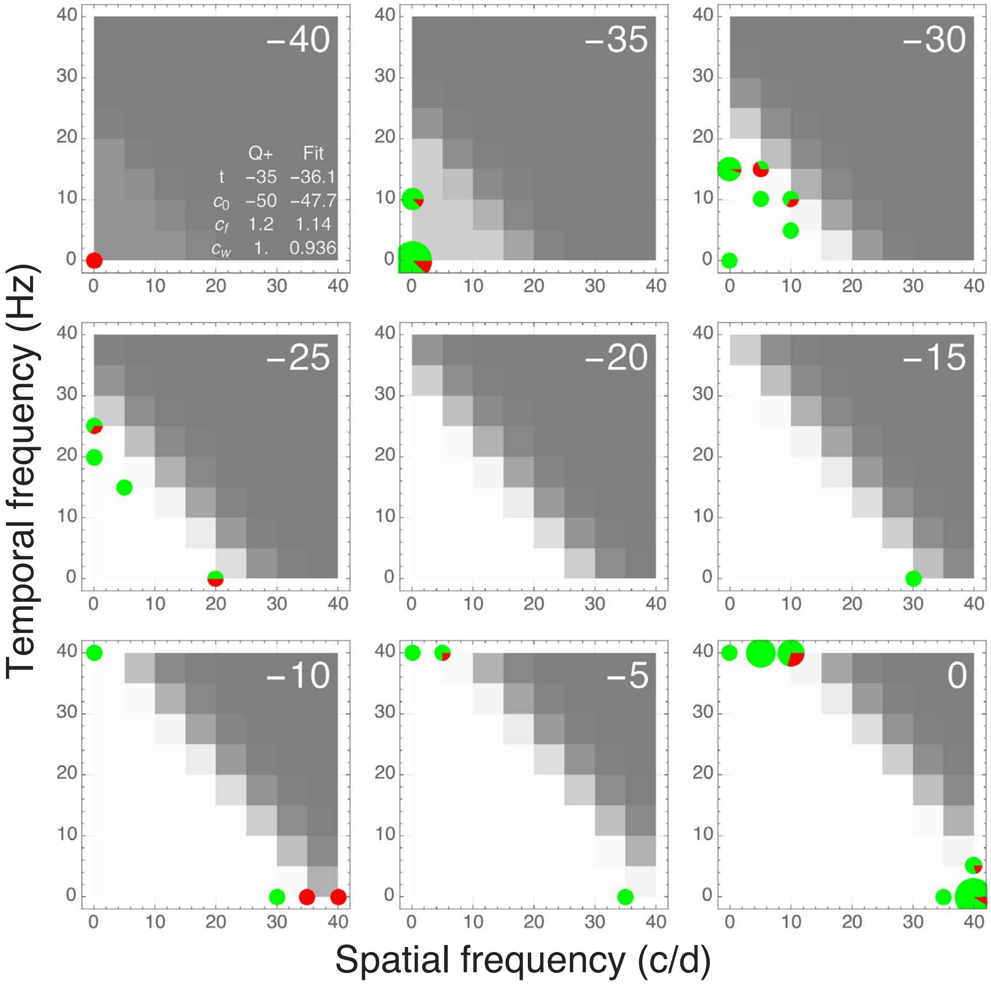


Figure 9. Placement of 64 simulated trials in measurement of the spatiotemporal contrast sensitivity function.

Downloaded from jov.arvojournals.org on 11/02/2020

In a recent abstract, Lesmes, Gepshtein, Lu, and Albright (2009) describe a different approach to this problem, using multiple coordinated two-dimensional slices through the three-dimensional stimulus space. With the QUEST+ method, no special treatment is required beyond expanding the number of stimulus dimensions to three.

Thurstone scaling {2, 3, 2}

This is a third example of multiple stimulus dimensions. Scaling experiments often rely on Thurstone's "Law of Comparative Judgment", in which stimuli varying along one physical dimension are assumed to generate internal responses along a one-dimensional internal perceptual scale (Thurstone, 1959). An example might be the relationship between luminance and brightness. The mapping between physical values x and perceptual values is given by a scaling function $\Theta(x)$.

Because of neural noise, the perceptual value varies from presentation to presentation of the same physical value. In Thurstone's case five, the internal distributions are assumed to be Normal with a fixed *SD* of 1. The internal scale is effectively measured in units of the standard deviation, so we can also consider it to express units of just noticeable difference or d'. Consider an experiment consisting of pair comparisons: asking the observer which is greater of two stimuli differing along the physical dimension. According to the Thurstone model, the probability of a correct judgment in a pair comparison is a function only of the distance between the sensory magnitudes induced by the two intensities of the pair

$$\Psi(\{x_1, x_2\}) = \Phi\left(\frac{\Theta(x_1) - \Theta(x_2)}{\sqrt{2}}\right) \quad (9)$$

where Φ is the cumulative Normal distribution.

We can therefore estimate the scale function by collecting many pair comparisons that span the range of possible values of x_1 and x_2. However, this is typically a daunting task because, if there are n discrete values along the x dimension and we need k trials at each to get reliable results, we will need $k\ n^2$ trials. And many trials will be uninformative because they involve separations that are much too easy to discriminate or much too hard. Previously, we designed an adaptive method to deal efficiently with this problem (Watson & Kreslake, 2001). Here we show that it fits within the QUEST+ methodology.

First, we adopt a model for the scaling function,

$$\Theta(x; m, t, q) = m(1-t)^{-q} \max(0, x-t)^q = m\left(\frac{\max(0, x-t)}{1-t}\right)^q \quad (10)$$

In this model, the physical scale is bounded by zero and one. The parameters are m, the maximum scale value, t, a threshold value of x below which the response is zero, and q, a power. This model is widely used in psychophysical scaling theory (Engen, 1972; Falmagne, 1986). An example is illustrated in Figure 10. The function is zero until the threshold is reached, and then it rises as a power function to the maximum at $x = 1$.

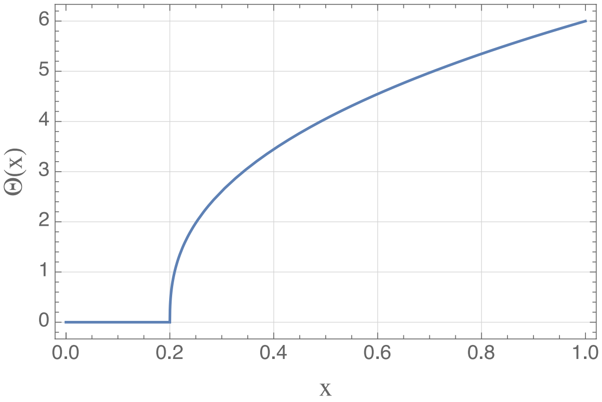


Figure 10. An example of a scaling function as defined by Equation 10 with parameters $m = 6$, $t = 0.2$, $q = 0.4$.

Downloaded from jov.arvojournals.org on 11/02/2020

Equations 9 and 10 lead directly to code for a scaling function and a psychometric function.

```
scale[x_, {max_, threshold_, power_}] := max (Max[0, x - threshold]/(1 - threshold))^power


QpPFScaling[x_List, p_List] :=
   {1 - #, #} & @ CDF[NormalDistribution[0, Sqrt[2]], -Subtract @@ (scale[#, p] & /@ x)]
```

For a simulated experiment, we set some options. We set the parameters of the simulated observer to those pictured in Figure 10. We establish modest sample ranges for the three parameters.

```
options=Sequence[
  QpOutcome-> (SimulatedObserver[#, QpPFScaling,{6,.2,.4}]& )
, QpXNames->{"x1","x2"}
, QpPNames->{"max","threshold","power"}
, QpXSamples->{Range[0,1,.1],Range[0,1,.1]}
, QpPSamples->{Range[1,10],Range[0,.9,.1],Range[.1,1,.1]}
, QpPsi-> QpPFScaling];
```

Now we run the experiment with 128 trials. If they were distributed equally over all pairs, there would be only about one trial per pair. We request an abbreviation of the printed output.

```
result = QuestPlus[128, options]//Short
```

```
{{6,0.2,0.4},{{{0.,0.7},2},<<126>>,{{0.2,0.3},2}}}
```

The quantized estimated parameters are close to those of the simulated observer. We can see where QUEST+ placed the trials by plotting them in the two-dimensional stimulus space. We plot the data on top of a gray-level representation of the two-dimensional psychometric function (Figure 11). Note that the placement of trials is concentrated near to the threshold and at $x_2 = 1$.

There is a symmetry to this experiment. We are asking whether x_2 is larger than x_1, but because both are values from the same dimension, we get the same information from a trial at $\{x_1, x_2\}$ as we get from $\{x_2, x_1\}$. Because we are only interested in deriving the parameters, it doesn't matter to us whether Quest+ chooses one or the other. Likewise, we could process the data in such a way that on all trials in which x_1 is greater than x_2, we exchange the stimulus values and exchange the responses.

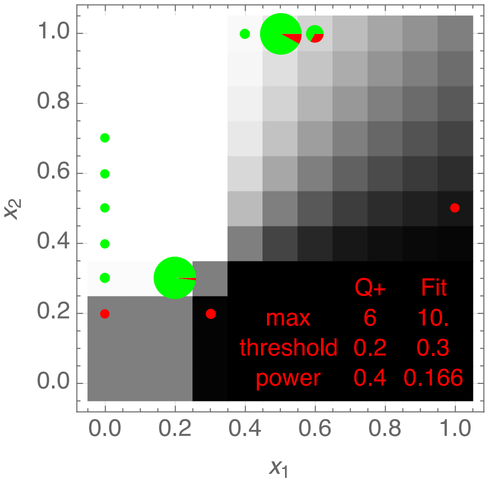


Figure 11. Simulated data from Thurstone scaling pair comparison experiment superimposed on the psychometric function for a simulated observer. Each symbol shows percentage correct as the green fraction, and the size reflects the number of trials. Quantized and fitted parameters are also shown.

Downloaded from jov.arvojournals.org on 11/02/2020

Blur discrimination {2, 2, 2}

This is another example of two stimulus dimensions that is also a template for a wide variety of “increment threshold” experiments. Here we simulate an experiment in which an observer is presented with two luminance edges and is asked to say which is more blurred. One edge is considered the reference with a specific blur. The other has an additional increment blur. The threshold increment is measured as a function of the reference blur. Watson and Ahumada (2011) reviewed many such experiments and showed that many of the resulting data sets are reasonably well modeled by a so-called Weber model, in which the just noticeable increment δ (arcmin) is given by

$$\delta = -\sigma + \sqrt{\beta^2(\omega^2 - 1) + \sigma^2\omega^2} \quad (11)$$

where σ is the reference blur (arcmin), β (arcmin) is an intrinsic “neural” blur, and ω is a parameter greater than one. An example of this function is shown in Figure 12 on log–log coordinates.

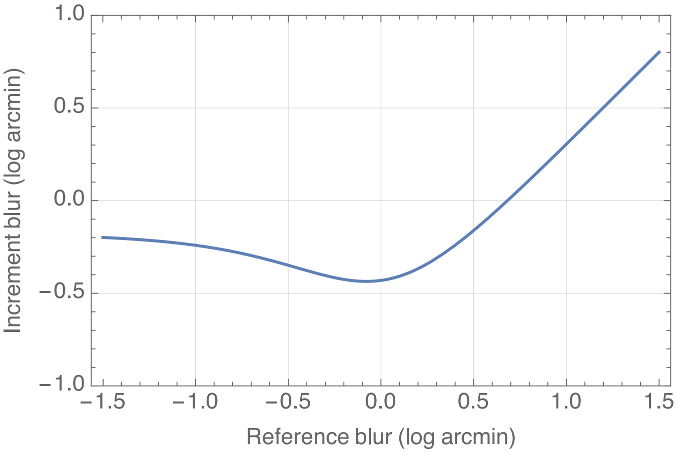


Figure 12. Threshold blur increment function of the Weber model (Watson & Ahumada, 2011).

We can construct a psychometric function of two stimulus dimensions for this case. We use log magnitudes for both reference and increment. We first write a function that duplicates Equation 11:

```
blurinc[sigma_, omega_, beta_] := -sigma + Sqrt[beta^2 (omega^2 - 1) + sigma^2 omega^2]
```

Then we code a suitable psychometric function, building it on the Weibull function. We use log values as the stimulus parameters. Notice also that we set the Weibull slope to 60. In the decibel domain with contrast thresholds, it is usually near to three. But here we are using log units, so the slope must be multiplied by 20.

```
QpPFBlur[{lsigma_, ldelta_}, {omega_, beta_}] :=
    QpPFWeibull[{ldelta}, {Log[10, blurinc[10^lsigma, omega, beta]], 60, .5, .01}]
```

We are now ready to define the options and simulate the experiment.

```
options = Sequence[
   QpOutcome -> (SimulatedObserver[#, QpPFBlur, {1.2, 1}] & )
  , QpPNames -> {"omega”, “beta"}
  , QpXNames -> {"Reference blur (log arcmin)”, “Threshold blur (log arcmin)"}
  , QpXSamples -> {Range[-1, 1.5, .25], Range[-1, 1, .25]}
  , QpPSamples -> {Range[1.05, 1.5, .05], Range[.5, 2, .1]}
  , QpPsi -> QpPFBlur];
```

Downloaded from jov.arvojournals.org on 11/02/2020

```
(result = QuestPlus[32,options])//Short
```

```
{{1.2,1.},{{{-0.25,-0.25},2},<<31>>}}
```

The quantized parameter estimates are reasonable. We can use our previous graphical methods to picture the results as shown in Figure 13. We see that nearly all the trials are placed at two locations: near to the "dipper" that serves to define beta and at the highest reference that serves to define omega.

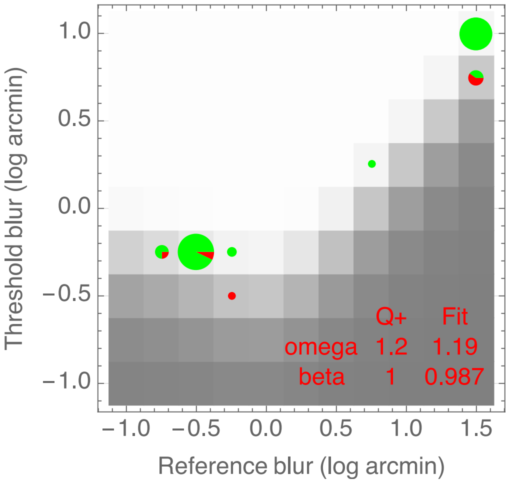


Figure 13. Simulated data from a blur increment experiment superimposed on a grayscale rendering of the psychometric function. Quantized and fitted parameters are also shown.

As noted above, this example serves as a template for many experiments, such as increment thresholds (Nachmias & Kocher, 1970), contrast masking (Foley, 1994; Legge & Foley; 1980), and noise masking (Lesmes et al., 2006; Pelli & Farell, 1999).

Categorical rating {1, 4, 4}

In this example, we illustrate a case of more than two responses. Consider a categorical rating scale in which stimuli are arrayed along a single dimension, and the observer must assign each stimulus to one of several categories. The categories could be integer numerical ratings as one example. This situation has been addressed with multiple QUEST staircases, one for each criterion (de Berker et al., 2016), but here we show it can be done with a single QUEST+ procedure.

We assume the observer has a set of internal criteria and assigns the stimulus to a particular category if the noisy internal response is greater than one criterion but less than the next. We design a psychometric function that has a list of parameters consisting of a standard deviation, followed by an arbitrary number of category widths. The number of possible responses is equal to the number of widths, plus one (the top category has no upper bound). This formulation is general and can accommodate an arbitrary number of criteria.

```
QpPFRating[{x_}, s_] := Block[{vals, crit},
 crit = Accumulate[Rest[s]];
 vals = CDF[NormalDistribution[#, First[s]], x] & /@ crit;
 vals = Append[-Differences[vals], Last[vals]];
 Prepend[vals, 1 - Total[vals]] ]
```

To keep the example simple, we consider the case of just four responses or three criteria. In Figure 14, we picture this case with a *SD* of 0.5 and category widths of {1, 2, 3}, meaning that the criteria are at one, three, and six. The psychometric function returns four probabilities, corresponding to the four responses. At any stimulus value, of course, the four curves must sum to one.

Downloaded from jov.arvojournals.org on 11/02/2020

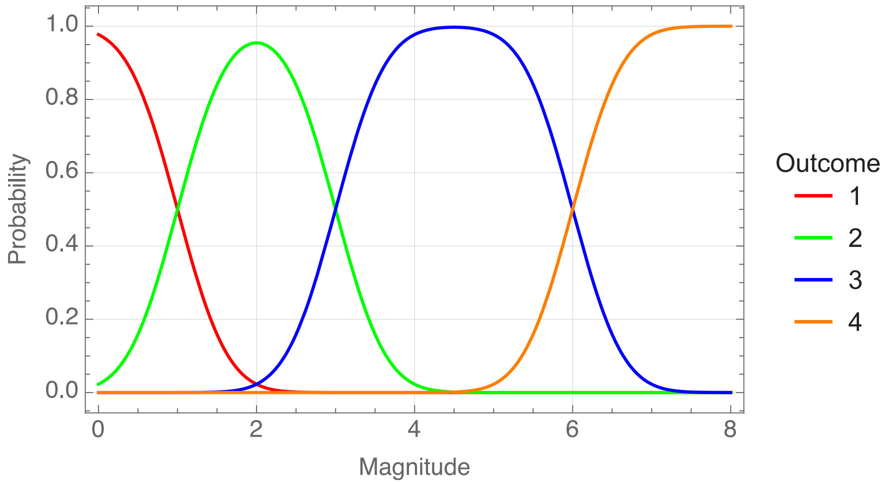


Figure 14. Psychometric function QpPFRating for a categorical rating experiment with four outcomes. The function parameters are {0.5, 1, 2, 3}.

We define the options for this experiment as follows:

```
options = Sequence[
  QpOutcome -> (SimulatedObserver[#, QpPFRating, {.5, 1, 2, 3}] & )
   , QpXNames -> {"Magnitude"}
   , QpPNames -> {"SD", "Width1", "Width2", "Width3"}
   , QpXSamples -> {Range[0, 9, .25]}
   , QpPSamples -> {Range[.05, .5, .05], Range[.5, 4, .5], Range[.5, 4, .5], Range[.5, 4, .5]}
   , QpPsi -> QpPFRating];
```

We then run 64 trials of the simulated experiment.

```
result = QuestPlus[128, options]//Short
```

```
{{0.45,1.,2.,3.},{{{5.25},3},{{2.75},2},<<124>>,{{2.75},2},{{2.25},2}}}
```

Note that the quantized estimated parameters are close to the simulated parameters. We plot the data in Figure 15 along with the fitted psychometric function, using the function **QpPlotCategories**. To allow us to show all four responses, each is shown with a different quarter circle symbol so that they do not overlap. As in Figures 1 through 4, the color saturation reflects the number of trials. The trial placement reflects the strategy of entropy minimization: Few trials are placed in the center of any category; rather they are placed near the boundaries (the criteria).

```
QpPlotCategories[result,options]
```

Downloaded from jov.arvojournals.org on 11/02/2020

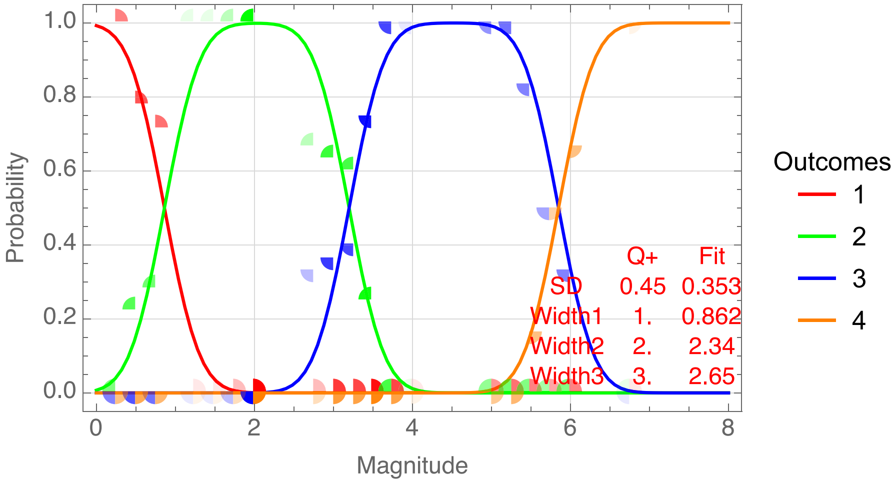


Figure 15. Simulation of 128 trials of a categorical rating experiment with four categories. The solid curves are the fitted psychometric function. The disc sectors are the percentage of each response. The saturation reflects the number of trials. Quantized and fitted parameters are also shown.

Categorization on a circular stimulus dimension {1, 2, 3}

Certain stimulus dimensions, such as direction and hue, are considered circular (Suchow, Brady, Fougnie, & Alvarez, 2013; Zhang & Luck, 2008). In such cases, we can again conduct a categorization experiment, but special care must be taken to deal with the circularity.

Here we consider a psychometric function for n categories on a circular dimension extending from 0 to 2π. We assume the internal response follows a von Mises distribution with parameters *mean* and *concentration* (Fisher, 1995). Concentration is similar to inverse variance, and when it is zero, the density is constant at $1/(2\pi)$. We assume the mean is given by the stimulus value. We further assume that the observer maintains $n - 1$ criteria that separate the circular dimension into n regions associated with the n possible responses.

We can construct a psychometric function QpPFCircular that will return the n probabilities. The list of parameters is {concentration, criterion$_1$, width$_1$, . . ., width$_{n-1}$}. Only $n - 1$ widths are specified because the n widths must total 2π. We can then configure QUEST+ with the following options:

```
options = Sequence[
   QpOutcome -> (SimulatedObserver[#, QpPFCircular, {4, Pi/3, 2 Pi/3, 2 Pi/3}] & )
  , QpNames -> {"Concentration", "c1", "w1", "w2"}
  , QpXNames -> {"Direction"}
  , QpXsamples -> {Range[0, 17] Pi/9}
  , QpPsamples -> {Range[1, 8], Range[0, 17] Pi/9, {2 Pi/3}, {2 Pi/3}}
  , QpPsi-> QpPFCircular];
```

This specifies a simulated observer with a concentration of four, a first criterion at $\pi/3$, and equal widths of $2\ \pi/3$. Because two widths are specified, there will be three possible outcomes. In the list of parameter samples, note that for each of the two width parameters we give only a single value. This means the parameter will not be estimated but fixed at that value. This is a convenient way of specifying a fixed parameter.

We can run the simulated experiment as follows (the //Short term suppresses lengthy printed output):

```
(result = QuestPlus[64, options])//Short

{{4,-π/3, 2π/3, 2π/3}, {{{-7 π/9},3},<<62>>,{{-<1>},3}}}
```

We see that QUEST+ produced perfect quantized estimates of the parameters. To see how QUEST+ placed the trials and what their outcomes were, we plot the data in Figure 16. Here again we have resorted to disc sectors to plot the proportion of the three responses at each stimulus value and have used saturation to indicate number of trials.

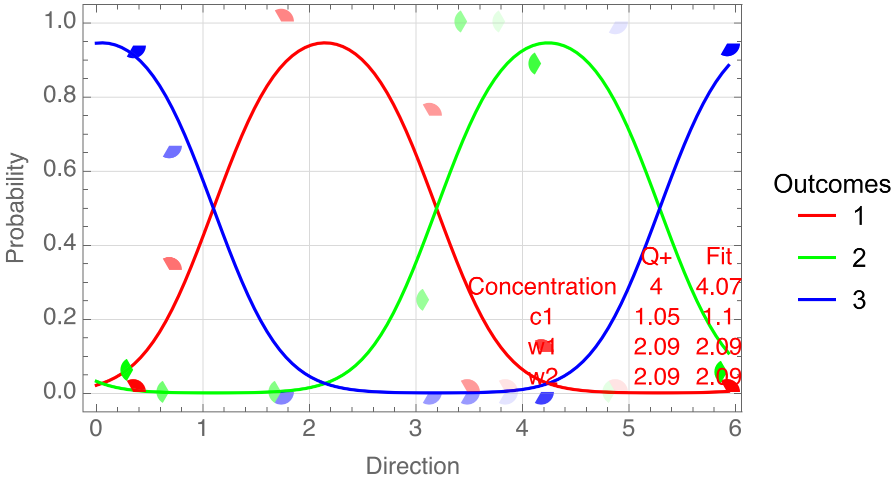


Figure 16. Simulation of QUEST+ experiment of categorization on a circular stimulus dimension.

Example code

One remarkable feature of the QUEST+ procedure is that, like QUEST, it can be implemented in a very few lines of code. In the original QUEST paper, we provided a brief program in the Basic programming language to make each step explicit and to illustrate the relative simplicity of the algorithm. We do the same here for QUEST+ but in the Mathematica language. In addition, we annotate the code to document what is happening in each step. However, in this case, rather than presenting a stripped-down example, we provide the actual complete code used to execute all of the examples shown in this paper. It should be emphasized that the code below works regardless of the number of stimulus dimensions, psychometric parameters, and trial outcomes. The few auxiliary functions required are provided in an appendix.

Define the default options.

```
Options[QuestPlus] = {
   QpOutcome -> (SimulatedObserver[#, QpPFWeibull, {-20, 3.5, .5, .02}] &),
   QpPNames -> {"Threshold (dB)”, "Slope", "Guess", "Lapse"},
   QpXNames -> {"Contrast (dB)"},
   QpXSamples -> {Range[-40, 0]},
   QpPSamples -> {Range[-40, 0], {3.5}, {.5}, {0.02}},
   QpPsi -> QpPFWeibull,
   QpPrior -> "Constant"};
```

The main function.

```
QuestPlus[ntrials_, options___Rule] ≔ Block[
```

Define the local variables.

```
{pnames, xsamples, psamples, psi, outcome, prior, response, data, depth, nxd, npd, nld, likelihood, likexprior,
   probability, posteriors, entropies, nextindex, xnext, posterior},
```

Set the values of the variables that can be set by options.

```
{xsamples, psamples, psi, response, prior} =
{ QpXSamples, QpPSamples, QpPsi, QpOutcome, QpPrior} /. {options} /. Options[QuestPlus];
```

Initialize the list of trials to an empty list.

```
data = {};
```

Compute the number of dimensions of the stimulus array.

```
nxd = Length[xsamples];
```

Compute the number of dimensions of the parameter array.

```
npd = Length[psamples];
```

Compute the number of dimensions of the likelihood array.

```
nld = nxd + npd + 1;
```

Compute the depth in the likelihood array at which parameter array begins.

```
depth = nxd + 1;
```

If necessary, define a constant prior. It is a uniform array with the same dimensions as the parameter array.

```
If[prior === “Constant”, prior = UniformArray[Length /@ psamples]];
```

Compute the likelihood array (conditional probabilities of each outcome at each stimulus value for each parameter value).

```
likelihood = Chop[N[Outer[psi[Take[{##}, nxd], Drop[{##}, nxd]] &,
   Sequence @@ Join[xsamples, psamples]]]];

likelihood = Transpose[likelihood, RotateLeft[Range[nld]]];
```

Set the posterior array equal to the prior array.

```
posterior = prior;
```

For each trial...

```
Table[(
```

Compute the product of likelihood and current posterior array at each outcome and stimulus location.

```
likexprior = Map[(# posterior) & , likelihood, {depth}];
```

Compute the total probability of each outcome at each stimulus location.

```
probability = Map[Total[#, Infinity] & , likexprior, {depth}];
```

Compute posterior PDF for each stimulus location and outcome.

```
posteriors = Map[unitize , likexprior, {depth}] ;
```

Compute the expected entropy for each stimulus location.

```
entropies = Total[MapThread[(ArrayEntropy[#1] #2) &, {posteriors, probability}, depth]];
```

Find the index of the stimulus with the smallest expected entropy.

```
nextindex = ListArgMin[entropies];
```

Set the next stimulus location to the location of the smallest expected entropy.

```
xnext = MapThread[#1[[#2]] &, {xsamples, nextindex}];
```

Do a trial of a simulated or real observer.

```
outcome = response[xnext];
```

Update the list of trials.

```
data = Append[data, {xnext, outcome}];
```

Set the new posterior to one of the previously computed posteriors, based on the outcome and stimulus location.

```
posterior = posteriors[[outcome, Sequence @@ nextindex]];
```

Repeat until the number of trials is complete.

```
), {ntrials}];
```

Estimate the optimal quantized Bayesian parameters by the location of the maximum in the final posterior.

```
{MapThread[#1[[#2]] &, {psamples, ListArgMax[posterior]}], data}]
```

Complexity and timing

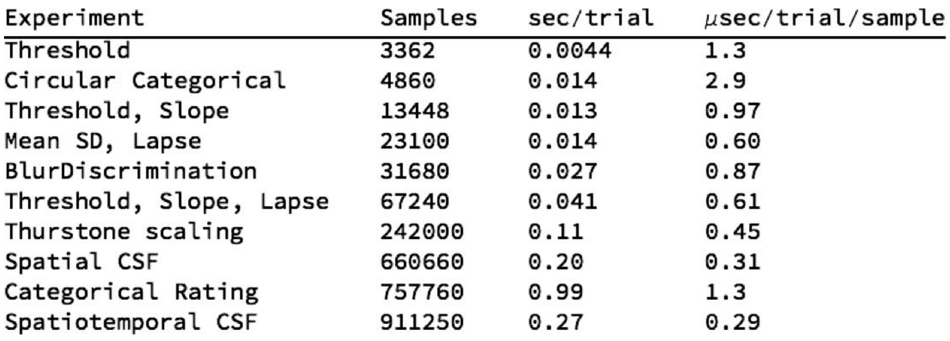
The QUEST+ procedure employs computations with and direct search through data structures of possibly high dimension. In particular, the likelihood array that represents the conditional probability of each outcome at each set of stimulus parameters for each set of psychometric function parameters can become very large. Before each trial, several calculations are performed with this structure. Its size therefore largely determines the time required to compute the optimal next stimulus position.

The size of the likelihood array is given by the number M,

$$M = N(\mathbf{r})N(\mathbf{x})N(\mathbf{s})$$

where N is an operator giving the number of samples in an array. This is just the product of the lengths of all the lists of responses, of stimulus samples, and parameter samples. Clearly, this number can become large rather quickly if care is not taken. However, on fast modern computers, this direct approach is quite practical for many problems. We have performed timing tests on the 10 examples discussed above, and the results show that the time per trial is constant for each experiment. The timings below were conducted in the relatively slow Mathematica language (Version 10.4) on a modest laptop (MacBook Pro, 2.5 GHz Intel Core i7 processor, 16 GB 1600 MHz DDR3 memory). We conducted simulations with 32, 64, and 128 trials and fit the resulting timings with a linear function of trials. In each case, the fit was nearly perfect. The slope of each fit is a measure of seconds required to compute the stimulus location for the next trial. These values are shown as s/trial in Table 1. These values are always less than 1 s/trial for the examples tested. We also divide s/trial by the number of samples in the likelihood array (M), and that is shown in Table 1 as μ s/trial/sample.

Downloaded from jov.arvojournals.org on 11/02/2020



Experiment	Samples	sec/trial	μsec/trial/sample
Threshold	3362	0.0044	1.3
Circular Categorical	4860	0.014	2.9
Threshold, Slope	13448	0.013	0.97
Mean SD, Lapse	23100	0.014	0.60
BlurDiscrimination	31680	0.027	0.87
Threshold, Slope, Lapse	67240	0.041	0.61
Thurstone scaling	242000	0.11	0.45
Spatial CSF	660660	0.20	0.31
Categorical Rating	757760	0.99	1.3
Spatiotemporal CSF	911250	0.27	0.29

Table 1. For the 10 example experiments, the number of samples in the likelihood array, the time to compute the next stimulus location, and the time per sample.

Given the relatively slow pace of psychophysics with human observers, all of the timings are within a practical range. The time per trial is a function of the number of dimensions (of responses, stimuli, and psychometric function) but also, of course, of the number of samples in each dimension. In the design of an experiment, it is important to keep the number of samples to a practical number. This can be done in part by noting that the sampling in a given dimension can often be quite coarse. QUEST+ is a method for data collection and provides only constrained, quantized parameter estimates. Those quantized estimates of parameter values returned by QUEST+ should always be refined by a subsequent unconstrained and unquantized fit to the data.

Simulations

We do not provide here any extensive simulations of the QUEST+ method because the variety of applications and configurations is essentially infinite. To reassure the reader, we do provide one simulation of a common configuration: estimation of threshold, slope, and lapse of a Weibull psychometric function in a two-alternative, forced choice experiment. Here the simulated observer obeyed a Weibull psychometric function with a threshold, slope, and lapse drawn from the same uniform distribution that specifies the prior with bounds of {–40, 0}, {2, 5}, and {0, 0.02}. Specifically, the options for all runs were

```
options = Sequence[
   QpXSamples -> {Range[-40, 0]}
  , QpPSamples -> {Range[-40, 0], Range[2, 5], Range[0, .02, .01]}
  , QpPsi -> (QpPFWeibull[#1, {#2[[1]], #2[[2]], .5, #2[[3]]}] &) ];
```

We completed 1,024 runs of 256 trials. We subsequently analyzed bias and standard deviation after completion of 32, 64, 128, and 256 trials. The results are shown in Figure 17. The results show that bias is small and approaches zero, and the standard deviation is also modest and declining, especially for the quantity that is usually of greatest interest: the threshold.

Downloaded from jov.arvojournals.org on 11/02/2020

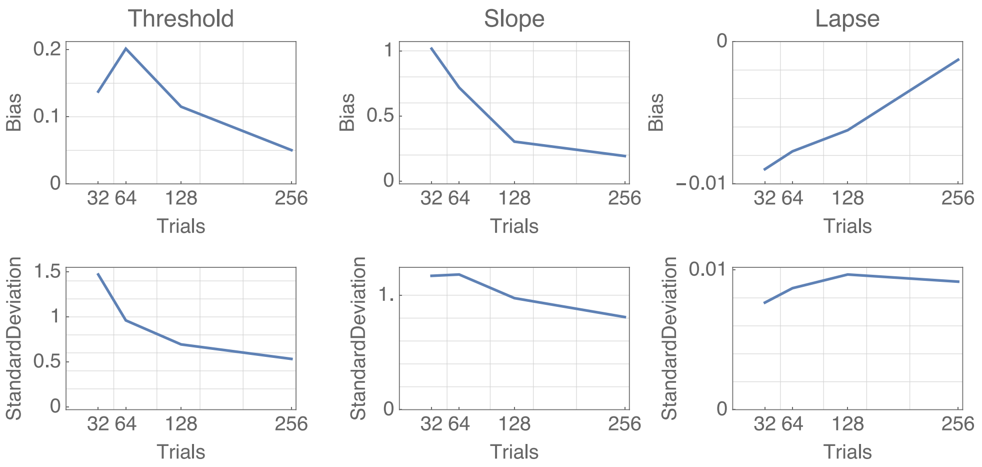


Figure 17. Simulations of QUEST+ estimation of threshold, slope, and lapse of a Weibull psychometric function.

Discussion

We have introduced an extension and generalization of the QUEST method. It accommodates an arbitrary number of stimulus dimensions, an arbitrary number of psychometric function parameters, and an arbitrary number of trial outcomes. We have shown how it can be applied to a variety of experiments, well beyond traditional threshold measurement.

The QUEST+ method is a parametric method. It requires an explicit parametric description of the psychometric function. It is best suited to cases in which previous results have revealed a common structure that can be described with a few parameters. As with all experimental design, some knowledge and art must be employed to establish the appropriate stimulus and parameter dimensions, the outcomes, and the psychometric function. For example, for the stimulus and parameter dimensions, one must decide on appropriate bounds, the density of samples, and whether they are best arranged in a linear sequence, a logarithmic sequence, or some other sequence altogether. But those are decisions that would need to be made even if a traditional inefficient exhaustive survey of the stimulus space were conducted.

Like all existing adaptive parametric psychometric methods, QUEST+ is a generalization and extension of previous methods. It is built upon the initial methods of QUEST (Watson & Pelli, 1979, 1983) but extended through the use of a minimum entropy placement rule (Kontsevich & Tyler, 1999), and generalized to multiple psychometric parameters (King-Smith & Rose, 1997; Kontsevich & Tyler, 1999; Snoeren & Puts, 1997), stimulus parameters (DiMattina, 2015; Kujala & Lukka, 2006; Lesmes et al., 2006; Lesmes et al., 2010), and multiple trial outcomes. Finally, these extensions and generalizations have been packaged in a single general software formulation that allows easy customization for arbitrary configurations. We have illustrated a number of configurations varying in number of stimulus parameters, psychometric parameters, and responses.

This generalized formulation of QUEST+ accommodates a remarkable variety of experimental configurations. The need for efficient data collection is general. It transcends the narrow confines of threshold measurement. It is our hope that researchers will recognize in this method how their own research can be accelerated.

Acknowledgments

This paper was written while the author was a visiting professor of psychology at New York University Abu Dhabi on leave from NASA Ames Research Center. I thank Kartik Sreenivasan for the simple question that got me started and the faculty, students, and postdocs of the NYUAD psychology department who listened politely to my first incoherent ramblings on this subject. I also thank NYUAD and in particular Dave Scicchitano for providing the time and resources to complete this project.

Commercial relationships: none.
Corresponding author: Andrew B. Watson.

Downloaded from jov.arvojournals.org on 11/02/2020

Email: abwatson@me.com.
Address: Los Gatos, CA, USA.

References

de Berker, A. O., Rutledge, R. B., Mathys, C., Marshall, L., Cross, G. F., Dolan, R. J., & Bestmann, S. (2016). Computations of uncertainty mediate acute stress responses in humans. *Nature Communications 7*, 10996.

DiMattina, C. (2015). Fast adaptive estimation of multidimensional psychometric functions. *Journal of Vision*, *15*(9):5, 1–20, doi:10.1167/15.9.5. [PubMed] [Article]

Emerson, P. L. (1986). Observations on maximum likelihood and Bayesian methods of forced-choice sequential threshold estimation. *Perception & Psychophysics*, *39*, 151–153.

Engen, T. (1972). Psychophysics II. Scaling methods. In J. W. Kling & L. A. Riggs (Eds.), *Woodworth and Schlosberg's Experimental Psychology* (pp. 47–86). New York: Holt, Rinehart and Winston.

Falmagne, J. C. (1986). Psychophysical measurement and theory. In K. Boff, L. Kaufman, & J. Thomas (Eds.), *Handbook of Perception and Human Performance* (pp. 1.3–1.66). New York: Wiley.

Fisher, N. I. (1995). *Statistical analysis of circular data*. Cambridge, UK: Cambridge University Press.

Foley, J. M. (1994). Human luminance pattern mechanisms: Masking experiments require a new model. *Journal of the Optical Society A*, *11*(6), 1710–1719.

Hall, J. L. (1968). Maximum-likelihood sequential procedure for estimation of psychometric functions [Abstract]. *Journal of the Acoustical Society of America*, *44*, 370.

King-Smith, P. E., Grigsby, S. S., Vingrys, A. J., Benes, S. C., & Supowit, A. (1994). Efficient and unbiased modifications of the QUEST threshold method: Theory, simulations, experimental evaluation, and practical implementation. *Vision Research*, *34*(7), 885–912.

King-Smith, P. E., & Rose, D. (1997). Principles of an adaptive method for measuring the slope of the psychometric function. *Vision Research*, *37*(12), 1595–1604.

Kontsevich, L. L., & Tyler, C. W. (1999). Bayesian adaptive estimation of psychometric slope and threshold. *Vision Research*, *39*(16), 2729–2737.

Kujala, J. V., & Lukka, T. J. (2006). Bayesian adaptive estimation: The next dimension. *Journal of Mathematical Psychology*, *50*(4), 369–389.

Legge, G. E., & Foley, J. M. (1980). Contrast masking in human vision. *Journal of the Optical Society of America*, *70*(12), 1458–1471.

Lesmes, L. A., Gepshtein, S., Lu, Z.-L., & Albright, T. (2009). Rapid estimation of the spatiotemporal contrast sensitivity surface. *Journal of Vision*, *9*(8):696, doi:10.1167/9.8.696. [Abstract]

Lesmes, L. A., Jeon, S.-T., Lu, Z.-L. & Dosher, B. A. (2006). Bayesian adaptive estimation of threshold versus contrast external noise functions: The quick TvC method. *Vision Research*, *46*(19), 3160–3176.

Lesmes, L. A., Lu, Z.-L., Baek, J., & Albright, T. D. (2010). Bayesian adaptive estimation of the contrast sensitivity function: The quick CSF method. *Journal of Vision*, *10*(3):17, 1–21, doi:10.1167/10.3.17. [PubMed] [Article]

Nachmias, J., & Kocher, E. C. (1970). Visual detection and discrimination of luminance increments. *Journal of the Optical Society of America*, *60*(3), 382–389.

Pelli, D. G. (1987). The ideal psychometric procedure. *Investigative Ophthalmology & Visual Science*, *28*(3, Suppl.), 366.

Pelli, D. G., & Farell, B. (1999). Why use noise? *Journal of the Optical Society of America A*, *16*(3), 647–653.

Skottun, B. C., Bradley, A., Sclar, G., Ohzawa, I., & Freeman, R. D. (1987). The effects of contrast on visual orientation and spatial frequency discrimination: A comparison of single cells and behavior. *Journal of Neurophysiology*, *57*(3), 773–786.

Snoeren, P. R., & Puts, M. J. (1997). Multiple parameter estimation in an adaptive psychometric method: MUEST, an extension of the QUEST method. *Journal of Mathematical Psychology*, *41*(4), 431–439.

Downloaded from jov.arvojournals.org on 11/02/2020

Suchow, J. W., Brady, T. F., Fougnie, D., & Alvarez, G. A. (2013). Modeling visual working memory with the MemToolbox. *Journal of Vision*, *13*(10):9, 1–8, doi:10.1167/13.10.9. [PubMed] [Article]

Thurstone, L. L. (1959). *The measurement of values*. Chicago: University of Chicago Press.

Treutwein, B. (1995). Adaptive psychophysical procedures. *Vision Research*, *35*(17), 2503–2522.

Vul, E., Bergsma, J., & MacLeod, D. I. A. (2010). Functional adaptive sequential testing. *Seeing and Perceiving*, *23*(5), 483–515.

Watson, A. B., & Ahumada, A. J. (2011). Blur clarified: A review and synthesis of blur discrimination. *Journal of Vision*, *11*(5):10, 1–23, doi:10.1167/11.5.10. [PubMed] [Article]

Watson, A. B., & Ahumada, A. J. (2016). The pyramid of visibility. *Electronic Imaging*, *2016*(16), 1–6, doi:10.2352/ISSN.2470-1173.2016.16.HVEI-102.

Watson, A. B., & Kreslake, L. (2001). Measurement of visual impairment scales for digital video. *Proceedings of the SPIE*, *4299*, 79–89.

Watson, A. B., & Pelli, D. G. (1979). The QUEST staircase procedure. *Applied Vision Assoc. Newsletter*, *14*, 6–7.

Watson, A. B., & Pelli, D. G. (1983). QUEST: A Bayesian adaptive psychometric method. *Perception & Psychophysics*, *33*(2), 113–120.

Zhang, W., & Luck, S. J. (2008). Discrete fixed-resolution representations in visual working memory. *Nature*, *453*(7192): 233–235.

Appendix: Auxiliary functions

The following are a few auxiliary functions that are needed to evaluate the QuestPlus function shown in the text.

1. Find the location of the smallest element in a multidimensional array.

```
ListArgMin[list_] := First[Position[list, Min[list]]]
```

2. Find the location of the largest element in a multidimensional array.

```
ListArgMax[list_] := First[Position[list, Max[list]]]
```

3. Create an array with the specified dimensions in which all elements are equal and sum to one.

```
UniformArray[dim_List] := ConstantArray[1./(Times @@ dim), dim]
```

4. Normalize a PDF. This is slightly complicated because we may occasionally have an input that is all zeros. We unitize it to a uniform distribution because that will have the maximum entropy.

```
unitize[v_] := Block[{total},
 total = Total[v, Infinity];
 If[total > 0
  , v/total
  , UniformArray[Dimensions[v]]]]
```

5. Compute the entropy of an array.

```
ArrayEntropy[array_, base_: 2] := -Total[(# Log[base, #]) & @ DeleteCases[Flatten[array], 0 | 0.]]
```

Downloaded from jov.arvojournals.org on 11/02/2020

Appendix: Mathematica Notebook

As a supplement, I provide a Mathematica Notebook, **QuestPlus.nb**, that contains the complete code required to run experiments using the QUEST+ routines as well as to fit the data and plot the results. Each function is documented, and examples of use are provided.

Appendix: Demonstration

I provide an **interactive demonstration** in which the user can run any one of the examples described earlier in this report. The demonstration requires installation of the free Wolfram CDF Player.

Downloaded from jov.arvojournals.org on 11/02/2020