TESTING CURVATURES OF LEARNING CURVES ON TRIAL AND BLOCK AVERAGE DATA

Denis Cousineau, Sébastien Hélie, Christine Lefebvre

Université de Montréal

Running head: Testing curvatures

For correspondence:

Denis Cousineau, Département de psychologie
Université de Montréal,
C. P. 6128, succ. Centre-ville
Montréal (Québec) H3C 3J7 CANADA

Office: (514) 343-7981, Fax: (514) 343-2285
E-mail: denis.cousineau@umontreal.ca

We would like to thank Dominic Charbonneau, Guy Lacroix, Serge Larochelle and an anonymous reviewer for their comments on an earlier version of this text. Part of this text was presented at the 11th Annual Meeting of the Brain, Behaviour and Cognitive Science Society, Québec city (2001). This research was supported by the *Fonds pour la formation de chercheurs et l'aide à la recherche, Établissement de nouveaux chercheurs*.

Request for reprint should be addressed to Denis Cousineau, Département de psychologie, Université de Montréal, C. P. 6128, succ. Centre-ville, Montréal (Québec) H3C 3J7, CANADA, or using e-mail at denis.cousineau@umontreal.ca.

Abstract

Many models offer different explanations of learning processes, some of them predicting equal learning rates between conditions. The simplest method to assess this equality is to evaluate the curvature parameter for each condition followed by a statistical test. However, this approach is highly dependant on the fitting procedure, which may come with built-in biases difficult to identify. Averaging the data per block of training would help reduce the biases, but averaging introduces a severe distortion on the curve that can no longer be fitted by the original function. In this text, we first demonstrate what is the distortion resulting from block averaging. The "block average" learning function can thus be used to extract parameters when the performance is averaged over blocks or sessions. The use of averages eliminates an important part of the noise present in the data and allows good recovery of the learning curve parameters. Equality of curvatures can be tested using a test of linear hypothesis. This method can be performed on trial data or block average data but it is more powerful with block average data.

Testing Curvatures of learning curves on trial and block average data

Many experiments involve training in a task. This is commonly done to reduce the variability that would arise from unskilled participants. In these cases, the experimenter is only interested in the final level of performance, often described by one or a few summary values (mean response times, standard deviation, percent correct, etc.). However, some studies are not simply interested in a snapshot but in the whole dynamic of performance over training (e. g., Logan, 1988, Rickard, 1997, Shiffrin and Schneider, 1977). Because of the large number of data involved, it is often convenient to summarize them in a curve: the learning curve (Newell and Rosenbloom, 1981, Heathcote, Brown and Mewhort, 2000).

Learning curves describe the evolution of performance over trials *t*. They are given by the following equation:

$$f(t) = a + b\, g(t) \qquad (1)$$

where *a* is the asymptote of the curve, and *b* the amplitude. These two scaling parameters act as boundaries since initial performance is given by the value *a* + *b* and the final performance is given by *a*.[1] The function *g*(*t*) describes the type of curvature present in the learning curve. As such, *g*(*t*) is called the core of the learning curve and is often a function of a third parameter, the learning rate parameter *c* (Paul , 1994).

The purpose of this text is not to decide which type of learning curves describes best the data. This issue is still highly controversial. When performance is measured by response times, many authors defended the power curve (Newell and Rosenbloom, 1981, Logan, 1988) . Its core function is given by $g_{PC}(t) = t^{-c}$. But Heathcote and his colleagues raised some concerns over the recent years (Heathcote, Brown and Mewhort, 2000). They suggested that the exponential curve, given by its core $g_{EX}(t) = e^{-c\,t}$, was as good a contender. Other learning curves have also been proposed over the years, such as the general power curve ($g_{GP}(t) = (t + d)^{-c}$ (Newell and Rosenbloom, 1981) which has a free parameter d to take into account learning prior to the beginning of the task (also see Cousineau, Goodman and Shiffrin, 2002). In the context of memory research, the retention curve measuring percent recalled as a function of time is also a function that fits the framework of Eq. 1 (Wixted, 1990).

Which core function is the correct one is not a resolved issue. In addition to this theoretical question however, there is a more empirical question about the curvature present in the performance. Curvature (or learning rate) is a measure of the speed at which performance reaches the asymptote. In the following, the curvature is quantified by the learning rate parameter c, assuming one type of curve (exponential, power, etc.).

Some theories predict that the stimuli to be learned will affect the curvature (reduction of information theories for example; see Haider and Frensch, 1996), whereas others predict that the stimuli will not affect curvatures but only the bounds a and b (such as strength theories; see Dumais, 1979). Logan’s Instance-based theory predicts that curvatures will be equal for the mean response times and their standard deviations. This same prediction also holds for the SSTS*, a subset of the serial self-terminating class of models in visual search (Cousineau and Larochelle, submitted). The aim of this paper is twofold. First, because mean performances are often used, we present a simple method to recover the parameters {*a, b, c*} out of averaged performance. Second, we present a method to test if the learning rates of two or more curves are equal. This method is applicable as soon as one type of core function is assumed. The core function can be any function that fits Eq. 1 and so avoids the above controversy.

Fitting averages

Most theories of learning assume that learning occurs on a trial-by-trial basis. Yet, raw data (called trial data hereafter) are usually very erratic, making the learning curve hard to see. To reduce the noise present in the data, researchers usually aggregate their data over blocks of trials using averages. However, Rickard (1997) pointed out that the curve of the block averages generally does not have the same core as the curve of the trial data (as shown below). Yet, this fact should not discourage the use of averaged data:we show in this section how to obtain the learning function of a curve averaged over blocks of successive trials. As will be shown, fitting a curve of averaged data is as easy as fitting trial data but allows the recovery of the right parameters more efficiently.

Averaging curves

In what follows, we define $f(t)$ as the trial learning curve function. $f(t)$ is a function of the trial number *t*, going from 1 to *T*. We want to know what is the learning curve equation when the data are averaged over blocks of training. Let us define $\bar{f}(n)$ as the block average function over block number *n* when trial data are averaged in blocks of *N* trials each ($N > 0$ is a constant). Thus, *n* goes from 1 to T / N (*T* is assumed to be a multiple of *N*). In order to simplify the problem, we first examine the core function of the block average curve. Let $\overline{g}(n)$ be the block average core function. By definition of the arithmetic mean, we have:

$$\overline{g}(n) = \frac{1}{N} \sum_{i=(n-1)N+1}^{nN} g(i) .$$

Where *i* indexes all the *N* trials in the n^{th} block. This equation generally cannot be simplified in the discrete case, but if *N* is large, we can solve it using a continuous approximation:

$$\overline{g}(n) \approx \frac{1}{N} \int_{(n-1)N}^{nN} g(x) dx . \qquad (2)$$

Equation (2) can be solved for many learning curves, yielding the equation of the block average core function.

Because a simple linear transformation relates the trial function and the core trial function, and because averages are not altered by such transformations, we can simply add the scaling parameters around the block average core function to obtain the full block average function:

$$\overline{f}(n) = a + b\,\overline{g}(n)$$

Scale invariant curves

A first question to ask is: Which functions remain of the same type after averaging? In other words, we want to know which functions are scale-invariant. This will answer Rickard's point noted at the beginning of this section. Two scale invariant functions are easily identified, the first one being trivial: the line ($f(t) = a + b\,t$) and the exponential curve ($f(t) = a + b\,e^{-c\,t}$).

The line is a degenerate curve since it has no curvature parameter. Its core function is simply $g_{LN}(t) = t$. The scaling parameter b represents the slope whereas a represents the intercept. By solving Eq. (2) on g_{LN}, using blocks of size N, we obtain $\overline{g}_{LN}(n) = Nn - N/2$. Thus, $\overline{f}(n) = a + b(Nn - N/2) = (a - bN/2) + (bN)n$. By substituting $a - bN/2 \rightarrow a$' and $bN \rightarrow b$', we obtain $\overline{f}(n) = a' + b'n$ and see that the block average core function is of the same type as the trial core function. One difference is that the slope is now steeper, because it is expressed using different units (blocks vs. trials).

Similarly, we show that the exponential curve is also scale invariant. Solving Eq. (2) on g_{EX}, we find that its block average core function is given by:

$$\overline{g}_{EX}(n) = \frac{e^{-cN(n-1)} - e^{-cNn}}{c\,N}.$$

Factorizing the exponential to isolate the dependant variable n, we obtain:

$$\overline{g}_{EX}(n) = \frac{(e^{cN} - 1)}{cN} e^{-cNn} \tag{3}$$

By substituting $\frac{(e^{cN} - 1)}{cN} \rightarrow b$' and $c\,N \rightarrow c$', we have $\overline{g}_{EX}(n) = b'e^{-c'n}$

Thus, we see that the block average function of an exponential curve is also an exponential curve. With the scaling parameters a and b, this is a three-parameter curve $\{a, b, c\}$ for a given block size N.

Scale dependant curves

The famous power curve is scale dependent since, as we show below, the core function is not functionally the same as the block average function. The core of the power function is given by $g_{PC}(t) = t^{-c}$. Averaging the function over blocks of size N using Eq. (2), we obtain:

$$\bar{g}_{PC}(n) = \frac{(N(n-1))^{-(c-1)} - (Nn)^{-(c-1)}}{(c-1)N}.$$

Note that $N\,(n\text{-}1)$ is the first trial of the n^{th} block and $N\,n$ is the last trial of that block.[2] We therefore substitute $N\,(n\text{-}1) \rightarrow n_F$ and $N\,n \rightarrow n_L$ to obtain:

$$\bar{g}_{PC}(n) = \frac{{n_F}^{-(c-1)} - {n_L}^{-(c-1)}}{(c-1)N}. \qquad (4)$$

Adding scaling parameters a and b as in Eq. (1), we see that $\bar{f}_{PC}(n)$ is a three-parameter curve defined by $\{a, b, c\}$ given a certain block size N. Therefore, it can be fitted to averaged data using n_F and n_L instead of the block number n with no more difficulty than fitting a power curve.

Equation (4) is a difference between two power curves (or more precisely, the same power curve at two different moments). Yet, the core is functionally different from a power curve's core function ($\bar{g}\,(x) \neq g\,(x)$). Thus, fitting block average data with the trial function should results in (i) poor fit, and (ii) non interpretable learning rate parameters.

As an example, in top part of Figure 1, we generated simulated response times (SRT) using a power curve over 400 trials using the parameters $\{a = 0, b = 350, c = 0.455\}$. As expected, a power curve fits perfectly the raw data, and a minimization algorithm (such as PASTIS, Cousineau and Larochelle, 1997) can recover the parameters almost perfectly (with a precision of ± 0.1%). In the bottom part of Figure 1, the SRT were averaged into 10 blocks of $N = 40$ trials. The dotted line shows the best-fitting power curve. As seen, the power curve shows systematic deviations (poor fit considering that there is no noise, r = 0.973) and the estimated parameters $\{\hat{a} = 0.00, \hat{b} = 94.2, \hat{c} = 0.633\}$ bear no resemblance with the true parameters. The dashed line shows the best-fitting block average power curve (Eq. 4). The fit is almost perfect, even though we introduced a continuous approximation. Further, the minimization algorithm recovered the parameters with a precision of ± 0.1%. This shows that in the absence of noise, fitting the block average curve on averaged data is not more difficult than fitting the simpler trial data curve on raw data.

Insert Figure 1 about here

The first part of the Appendix explores the efficiency of the block average function to recover the parameters when noise is present. It shows that in general (trial data or block average data), the major factor that makes parameters difficult to recover is noise. The impact of noise can be reduced significantly by increasing the number of trials. The second part of the Appendix shows that it is preferable to use block averages when fitting parameters if the curvature is steep (*c* bigger than 0.4).

Illustrating the core function

One convenient way to look at curvature is to have a graph of the core function. Remember that all core functions starts at one and have an asymptote of zero. Thus, if the curves have an equal learning rate, their core functions should superimpose. Further, if block average data are plotted, standard error intervals (*SE*) around the core functions can be computed.

In order to plot the core function, one must first choose which curve is assumed to underlie the data. For example, it can be the power functions in the case of trial data, or Equations (4) if block average data are used. Isolating the core of a learning curve requires that each point at time *x* (trial or block number) be transformed using:

$$\hat{g}(x) = \frac{\hat{f}(x) - \hat{a}}{\hat{b}} \qquad (5)$$

where $\hat{a}$ and $\hat{b}$ are estimates of the two scaling parameters $\{a, b\}$ and $\hat{f}$ is the observed performance at time *x*. If both $\hat{a}$ and $\hat{b}$ are valid estimators, Eq. (5) returns a valid approximation of the core function $\hat{g}$.

If summary values are plotted (such as mean or standard deviation), the standard error intervals (*SE*) can be computed (this approach cannot be used with trial data). *SE* can be used as a general indicator whether two curves superimpose or not.

SE of the block average data at block *n* is given by $SE_{\overline{f}}(n) = \frac{\overleftrightarrow{f}(n)}{\sqrt{N}}$ where *N* is the number of observations per block and $\overleftrightarrow{f}(n)$ is the estimated standard deviation at block *n* (Cramér, 1946). Equation (5) requires *SE* for transformed scores but manipulating *SE* is well established (Tremblay and Chassé, 1970). For example, adding a constant to a score does not alter its standard error interval while multiplying it by a constant multiplies its standard error interval. The final block average core function is thus given by:

$$\hat{\hat{g}}(n) = \frac{\overline{f}(n) - \hat{a}}{\hat{b}} \pm \frac{\overleftrightarrow{f}(n)}{\hat{b}\sqrt{N}}$$

where $\overline{f}\,(n)$ and $\overleftrightarrow{f}(n)$ are the average and the standard deviation of the empirical measures at block *n*. Equivalent manipulations can be performed for any summary value normalized according to Eq. (5), as long as its standard error is known (Kendall and Stuart, 1983).

Illustrating the core function might provide an interesting solution to the related question: did performance reached the asymptote? Formally, the performance will never reach asymptote since for most learning curves it requires an infinite amount of practice. Nevertheless, subjects may reach a level where performance does not significantly differ from asymptotic performance. A very stringent criterion could be to declare *a priori* that asymptotic performances are reached if the core function is within 2 *SE* of zero on the last 4 blocks.

Testing curvatures

We describe in this section a method to test whether two or more curvatures are equal, irrespective of the scaling parameters (amplitude and asymptote). Consider the following curves, $f_1, f_2, \ldots, f_s$ with unknown parameters $\{a_i, b_i, c_i\}$ for the i^{th} curve. The most intuitive method to test that the curvatures are equal would consist in estimating the curvatures $\hat{c}_i$ (using a minimization procedure) and comparing them using a statistical test. However, this method has a very low power because it looses a lot of information (a large data set is compressed into a single estimate $\hat{c}_i$). Considering that in general experiments involving learning have only a few subjects, this compression is too drastic.

The test of linear hypothesis (Rao, 1959) avoids this problem because it constrains the fit on more than singleton c_i. Suppose that *s* data sets are available, forming *s* learning curves labeled f_1 to f_s. If the core functions g_i are all identical, then we can write:

$$\begin{gathered} f_1(t) = a_1 + b_1 g(t) \\ f_2(t) = a_2 + b_2 g(t) \\ \ldots \\ f_s(t) = a_s + b_s g(t) \end{gathered}$$

As a consequence, we can show that the average curve $f_{\bar{s}}$ is given by the average parameters and the core function::

$$f_{\bar{s}}(t) = E(a_i) + E(b_i) g(t)$$

where $E(a_i)$ is the average of the a_i and $E(b_i)$, the average of the b_i, $i = 1 .. s$. If the average curve $f_{\bar{s}}$ does not capture the data, it means that the core function is not unique to the *s* data sets. This is called a linear hypothesis.

One method to test that the curve with averaged parameters captures the average data set is the linear hypothesis test created by Rao in 1959. It has been mentioned in Paul (1994) but with minimal details. One objective of this section is to detail the structure of the test and to provide a short Mathematica listing that performs it. However, the real contribution of this section is to use the block average learning curve in conjunction with Rao's test and to show that doing so drastically increases the power of the test.

Applying the test of linear hypothesis to trial data

Following Rao (1959), the first step is to describe the model underlying the data set. In terms of vectors, let the model **M** = {1, *g(t)* } and the parameters θ = {α, β } so that $\theta^T\mathbf{M} = \alpha + \beta g(t)$. It must be understood that *g(t)* is also a function of *c*, the learning rate parameter. Suppose we have collected for each of the *s* data sets a number *T* of trial observations. The model **M** varies according to the trial number. The matrix **A** summarizes the evolution of the model for each trial and each parameter. We can write:

$$\mathbf{A} = \begin{pmatrix} 1 & g(1) \\ 1 & g(2) \\ & \cdots \\ 1 & g(T) \end{pmatrix}$$

where the first column indicates the contribution of α to the performance of the average curve and the second column, the contribution of β.

In the implementation of the model, c is not considered a parameter. Therefore, it must receive a value at this point. However, under the null hypothesis, every set has the same curvature, and the average curve is also representative of the curvature. Thus, a numerical value for c should be obtained using a least square minimization routine (such as PASTIS, Cousineau and Larochelle, 1997) on the between-set average data.

The next step is to obtain the set of estimates $\hat{\boldsymbol{\theta}}$ that offers the best fit. Rao (1959) proposed one method.[3] It is our experience that a better approach (less biased) is to take advantage of the null hypothesis that says that the group best-fitting parameters $\{\hat{\alpha}, \hat{\beta}\}$ ought to be the average of the individual subject best-fitting parameters. So let $\hat{\boldsymbol{\theta}}$ = $\{\hat{\alpha} = E(\hat{a}_i), \hat{\beta} = E(\hat{b}_i)\}$. In summary, (i) fit the average curve to obtain the curvature c, (ii) fit the individual and average the individual asymptotes and amplitudes to obtain the parameter set $\hat{\boldsymbol{\theta}}$. The estimate $\hat{\boldsymbol{\theta}}$ is only valid if the null hypothesis is not rejected.

In order to perform a statistical test, summary values are needed. The first summary value is a vector $\mathbf{y} = \{E(f_i(1)),\ldots,E(f_i(T))\}$ containing the between subject average performance for the various trials from 1 to T. The second summary value is a variance-covariance matrix (of size $T \times T$) called hereafter **S** such that:

$$\mathbf{S} = \begin{pmatrix} Var(f_i(1)) & Cov(f_i(1), f_i(2)) & \cdots & Cov(f_i(1), f_i(T)) \\ Cov(f_i(2), f_i(1)) & Var(f_i(2)) & & \\ \vdots & & \ddots & \\ Cov(f_i(T), f_i(1)) & & & Var(f_i(T)) \end{pmatrix}$$

where Var($f_i(j)$ *)* is the unbiased variance of the performances at time *j* and Cov($f_i(j)$, $f_i(k)$) is the unbiased covariance of the observations between trials at time *j* and trials at time *k*. This matrix is symmetrical.

The following equation is used to test the significance of the linear hypothesis. Let *r* be the number of data point in each of the curve *T* minus the number of parameters (generally three) and *n* the number of data set *s*. The test is of the form:

Reject H$_0$ if:

$$\mathcal{F} = \frac{n-r}{r} \times (\mathbf{y} - \mathbf{A}\hat{\boldsymbol{\theta}}) \mathbf{S}^{-1} (\mathbf{y} - \mathbf{A}\hat{\boldsymbol{\theta}}) > F(\alpha, r, n-r)$$

where $F(\alpha, r, n-r)$, the critical value for the decision at level *α*, say 5%, is read on a Fisher *F* table with *r, n - r* degrees of freedom for the numerator and the denominator respectively. In case where the inverse cannot be found (**S** is singular), a pseudo inverse can be used (Rao, 1959).

Overall, Rao's test of linear hypothesis requires (a) the type of learning function to fit, (b) a minimization procedure to find the group curvature and the individual asymptotes and amplitudes, (c) summary values (a vector of mean performance at trial $t \leq T$ and a $T \times T$ variance-covariance matrix), and (d) extensive matrix manipulation capabilities. This last point used to be the most difficult to obtain. Rao (1959) described a complex method to make optimal use of the desk calculator available at that time (to the point that the article is difficult to decipher). Schneiderman and Kowalski (1985) described an implementation of the test using SAS. Yet, this program is still difficult to follow. I present in Listing 1 a short Mathematica program to compute the summary values (**y** and **S**), the best fitting parameter $\hat{\boldsymbol{\theta}}$, and the statistic $\mathcal{F}$.

Insert Listing 1 about here

This approach is more powerful than the intuitive ones described at the beginning of the section because it does not reduce the data to a single value (*c* or r^2). In fact, when testing the hypothesis, all the points along the curves are used as constraints to see if the instantiated model **A** is capturing the individual observations.

As seen from the degrees of liberty, the test of linear hypothesis requires that the number of data set (generally, the number of subjects) be at least equal to the number of trials. Because a typical experiment often involves hundreds of trials, the number of subjects rapidly becomes prohibitive. As shown next, by collapsing the trial data into a fewer number of blocks, it allows measuring a smaller number of subjects and still have a powerful test.

Applying the test of linear hypothesis to block average data

First, we note that after performing block averaging, the *s* data sets now form *s* curves $\overline{f_i}$. These block average curves are not of the same type as the trial curves (unless they are scale invariant functions). However, their core functions $\overline{g}_i(n)$ are known (for examples, it is Eq. 4 in the case of a power curve). As such, under the null hypothesis that the curvatures are the same, we can write:

$$\begin{aligned}\overline{f_1}(n) &= a_1 + b_1\,\overline{g}(n)\\ \overline{f_2}(n) &= a_2 + b_2\,\overline{g}(n)\\ &\ldots\\ \overline{f_s}(n) &= a_s + b_s\,\overline{g}(n)\end{aligned}$$

Here, a_i and b_i are exactly the same as with the trial data. Thus, if all the curves have the same curvature (same core), we can also write:

$$\overline{f_{\bar{s}}}(n) = E(a_i) + E(b_i)\overline{g}(n)$$

where $\overline{f_{\bar{s}}}$ is the average across data sets of the block averages. Here, we have two distinct averaging: First, within data set to obtain the block average curves; next, between the block average curves to obtain a single $\overline{f_{\bar{s}}}$ curve. Also note that the relation between the block average curves ($\overline{f_{\bar{s}}}$ vs. the various $\overline{f_i}$) is the same as the relation between the trial data curves ($f_{\bar{s}}$ vs. the various f_i), one of a linear relationship. Hence, the test of linear hypothesis is relevant here for the same reasons it was for the trial data.

The model is $\overline{\mathbf{M}}$ = {1, $\overline{g}(n)$ } with parameters θ = {α, β} from which we can create the matrix $\overline{\mathbf{A}}$ instanciating the model.

As an example, if we assume that the trial data follow a power curve, the instantiation for block average data, following Eq. (4), is composed of lines for each block n of the sort $\left\{1, \frac{n_F{}^{-(c-1)} - n_L{}^{-(c-1)}}{(c-1)N}\right\}$ where c must be determined using least square methods, N is the number of trials per blocks, and n_F is the first trial of block n (given by $N \times (n\text{-}1)$) and n_L is the last trial of block n (given by $N \times n$). If there are T observations in the trial data sets, there are $T\,/\,N$ blocks in the block average data sets. Thus, the final matrix $\overline{\mathbf{A}}$ could be:

$$\overline{A} = \begin{pmatrix} 1 & \dfrac{(0N)^{-(c-1)} - (1N)^{-(c-1)}}{(c-1)N} \\ 1 & \dfrac{(1N)^{-(c-1)} - (2N)^{-(c-1)}}{(c-1)N} \\ & \text{ććć} \\ 1 & \dfrac{(T/N-1)^{-(c-1)} - (T\,/\,N)^{-(c-1)}}{(c-1)N} \end{pmatrix}$$

The matrix $\overline{\mathbf{A}}$ may look quite cumbersome. Yet, given c and N, it is easy to compute. In addition, it is now N times shorter, speeding up the remaining computations by a factor N.

Whether we fit the trial data using the trial function or the block average data using the block average function, the best fitting parameters $\hat{\boldsymbol{\theta}}$ should be identical. However, reducing the number of points tested using blocks makes it possible to measure a reasonable number of subjects. This would suggest that having very few blocks containing a lot of trials each is desirable (so that few subjects are required). This is not true: There is a trade-off between blocks of increasing size and power. At some point, the blocks are so large that there is only a few blocks left. A reasonable compromise is to choose a block size N less or equal to the square root of the total number of trials. The third section of the Appendix tests this claim with Monte Carlo simulations.

Discussion

The advantages of fitting average curves are numerous: the average data are less noisy than the trial data. It is therefore possible that the parameters $\{\hat{a}, \hat{b}, \hat{c}\}$ estimated from the averaged data will be more accurate (as shown in the Appendix). Further, the block average function $\bar{f}(n)$ is not more complex or more difficult to fit using a minimization algorithm (and Eq. 3 or 4). In particular, it has exactly the same number of free parameters. We updated the learning curve estimation program PASTIS to fit the block average functions (source code available at http://mapageweb.umontreal.ca/cousined/papers/02-pastis). However, it still requires that the modeler make an assumption about which type of curves (power, exponential, or other) underlies the data. Finally, when using block average data, standard errors can be computed around the core function.

The form of averaging presented here is a within-subject average. As shown by Estes (1956), between-subject averaging is risky if the individual subjects have different learning rates c. Indeed, the average of $f_1, f_2, \ldots, f_s$ cannot be solved unless the individual cs are known or are all equal. We presented in the second section a test of curvature based on Rao's test of linear hypothesis which can be use in order to solve this problem..

References

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

Cousineau, D. & Larochelle, S. (1997). PASTIS: A Program for Curve and Distribution Analyses. Behavior Research Methods, Instruments, & Computers, 29: 542-548.

Cousineau, D. & Larochelle, S. (submitted). Visual-Memory search: An integrative perspective. Psychological Review, : 0-0.

Cousineau, D., Goodman, V. & Shiffrin, R. M. (2002). Extending statistics of extremes to distributions varying on position and scale, and implication for race models. Journal of Mathematical Psychology, 46: 431-454.

Cramér, H. (1946). Mathematical Methods of Statistics. Princeton: Princeton University Press.

Dumais, S. T. (1979). Perceptual Learning in Automatic Detection: Processes and Mechanisms. Bloomington: unpublised Ph. D. presented at the Indiana University.

Estes, W. K. (1956). The problem of inference from curves based on group data. Psychological Bulletin, 53: 134-140.

Haider, H. & Frensch, P. A. (1996). The role of information reduction in skill acquisition. Cognitive Psychology, 30: 304-337.

Heathcote, A., Brown, S. & Mewhort, D. J. K. (2000). The power law repealled: The case for an exponential law of practice. Psychonomic Bulletin & Review, 7: 185-207.

Hoel, P.G (1964). Methods for comparing growth type curves. Biometrics, 20: 859-872.

Kendall, M.G. & Stuart, A. (1983). The advanced theory of statistics. London: C. Griffin.

Logan, G. D. (1988). Toward an instance theory of automatization. Psychological Review, 95: 492-527.

Newell, A. & Rosenbloom, P. S. (1981). Mechanisms of skill acquisition and the law of practice, in Anderson, J. R. (eds.). Cognitive Skills and their Acquisition (pp. 1-55). Hillsdale, NJ: Laurence Erlbaum Associates.

Paul, L.M. (1994). Making interpretable forgetting comparisons: Explicit versus hidden assumptions. Journal of Experimental Psychology: Learning, Memory and Cognition, 20: 992-999.

Rao, C.R (1959). Some problems involving linear hypotheses in multivariate analysis. Biometrika, 46: 49-58.

Rickard, T. C. (1997). Bending the power law: a CMPL theory of strategy shitfs and the automatization of cognitive skills. Journal of Experimental Psychology: General, 126: 288-311.

Schneiderman, E.D. & Kowalski, C.J. (1985). Implementation of Rao's one-sample polynomial growth curve model using SAS. American Journal of Physical Anthropology, 67: 323-333.

Shiffrin, R. M. & Schneider, W. (1977). Controlled and automatic human information processing: II Perceptual learning, automatic attending, and a general theory. Psychological Review, 84: 127-190.

Tremblay, L.-M. & Chassé, Y. (1970). Introduction à la Méthode Expérimentale. Montréal: Centre Educatif et Culturel inc.

Wixted, J.T. (1990). Analyzing the empirical course of forgetting. Journal of Experimental Psychology: Learning, Memory and Cognition, 16: 927-935.

Footnotes

[1] The point where the initial performance is measured depends on the type of curve. For the exponential curve, it is measured at time $t = 0$ and for the power curve, at time $t = 1$.

[2] Actually, $N\,(n - 1)$ returns zero as the first trial. For the power curve, it is inappropriate since, according to this type of curve, the performance is infinite at time $t = 0$. To solve this issue, we used $N\,(n - 1) + ½$ and $N\,n + ½$ when doing actual fitting. Thus, blocks are ranging from ½ to $N + ½$, $N + ½$ to $2N + ½$, etc.

[3] With the model implementation A and the summary values $\mathbf{y}$ and $\mathbf{S}$ (see next), we can obtain the optimal parameter $\hat{\boldsymbol{\theta}}$ for the group by solving $\hat{\boldsymbol{\theta}} = (\mathbf{A}^T\mathbf{S}^{-1}\mathbf{A})^{-1}\mathbf{A}^T\mathbf{S}^{-1}\mathbf{y}$ which yields the least mean square solution to the problem (Bates and Watts, 1988). This method is based on the postulate that the difference between the subjects remains the same with practice. It is not the case since between-subject variability diminishes with training (Cousineau and Larochelle, submitted).

Appendix: Fitting and testing curves using trial and block average data

The general objectives of this paper are to describe a method to estimate curvatures and test them. These objectives are crucially dependent on a minimization algorithm that reduces the sum of square error (SSE) between the data and the ideal curve passing through the points. The parameters $\hat{\boldsymbol{\theta}} = \{\hat{a}, \hat{b}, \hat{c}\}$ that minimizes the sum of square error are termed the best-fitting parameters.

Simulation 1: Testing biases using trial data

To explore the capabilities of a minimization algorithm to estimate the true parameters θ, we ran Monte Carlo simulations. We used the minimization software PASTIS (Cousineau and Larochelle, 1997) but we also tested the minimization procedure *FindMinimum* implemented in Mathematica and found no differences in the pattern of results. We present the results using the following measures of bias: the average distance between the i^{th} estimates $\hat{\boldsymbol{\theta}}$ and the true parameters θ, obtained over a large number of replications. Bias can also be seen as the distance between the center of gravity of all the $\hat{\boldsymbol{\theta}}_{\iota}$ and the true θ (i = 1 .. R, the number of replications):

$$Bias := \left\| E\left(\hat{\boldsymbol{\theta}}_i - \boldsymbol{\theta}\right) \right\| = \left\| E(\hat{\boldsymbol{\theta}}_i) - \boldsymbol{\theta} \right\|$$

where || $x - y$ || denotes the Euclidian distance between x and y in a 3D space. To express the bias as a percentage, we divided this value by || θ ||. In addition, we computed the efficiency, a measure of dispersion around the true parameters θ:

$$Efficiency := SD\left(\left\| \hat{\boldsymbol{\theta}}_i - \boldsymbol{\theta} \right\|\right) = \frac{1}{R-1} \sum_{i=1}^{R} \left\| \hat{\boldsymbol{\theta}}_i - \boldsymbol{\theta} \right\|^2$$

We generated power curve trial data. We kept constant the true asymptote at a = 300 and the true amplitude at b = 1000. Because they are linear parameters, they would bring little information if they were varied. However, we varied the learning rate because curves with almost non-existent curvatures might be more difficult to fit than curves with pronounced descent. We use $c = \{0.2, 0.4, 0.6, 0.8\}$. We also added a small amount of noise to the generated curves. We used normal additive noise with zero mean and standard deviation η times the height of the curve minus the asymptote. The values η used were 0.5, 1.0 and 2.0. A value of 2.0 represents a large variability that is similar to typical human RT data. At ∞, the curve would reach the asymptote (height of zero) and so noise would be zero, but of course, we never generated that many points. The number of points generated (sample size) was varied {50, 100, 200, 400, 800, 1600}. Each point represents one trial, starting at trial 1. Table A.1, column 2 recapitulates the factors.

Insert Table A.1 about here

For a given combination of curvature × sample size × noise, we generated a noisy curve and ran a minimization algorithm (PASTIS) to obtain the best-fitting parameters. We replicated this a thousand times, after which bias and efficiency were computed.

The results are shown in Figure A.1. As seen, noise had an important impact on bias and efficiency. The more noise, the less accurate are the best-fitting parameters. It is still a reasonably small bias on average since a typical set of estimated parameter is rarely more than 2% inaccurate. Sample size also had an important impact. Larger sample sizes tend to be less biased. Finally, the learning rates (small vs. steep) had no influence on the best-fitting parameters.

Insert Figure A.1 about here

Simulation 2: Testing biases using block average data

The above simulations were performed using trial data. Next, we want to see if there is an improvement in the best-fitting parameters when we use block average data instead of trial data. We ran a second series of simulations where we used both trial data and block average data. The size of a block *N* was 5, 10, 20, 40, or 80 trials per block. To keep the number of results manageable, we fixed the sample size at 400 trials. This implies a kind of trade-off since, as a consequence, the larger the block size *N*, the less points remain for fitting. Everything else is as in the previous simulations. The third column of Table A.1 recapitulates the fixed and varied factors.

The results are shown in Figure A.2. As can be seen, for $c = 0.8$ (bottom row), using blocks of increasing size reduces the bias and the efficiency. In the best case, bias is reduced twofold and efficiency by almost 50% (block size *N* of 80). Thus, even though there is only 5 points (400 trial data averaged by blocks of 80 trials), the parameters are recovered very accurately. However, this trend reversed for curvatures smaller than 0.5 where averaged data returns more biased and less efficient estimates. Thus, for small curvatures, the small amount of blocks (5, 10 and 20 – blocks of 80, 40 and 20 trials respectively) is very detrimental. In this case, the modeler should avoid estimating parameters on block average data.

Insert Figure A.2 about here

Simulation 3: curvature testing

We explored the reliability of the test of linear hypothesis. In order to perform a statistical test, we first generated 100 trial data sets following a power curve. As before, parameters *a* was fixed at 300 and *b* at 1000. Parameter *c* varied for each half of the sets with possible values of {0.2, 0.4, 0.6, 0.8}. When the two *c* where equal, the test should not reject H_o or else it makes a type-I error. When the two *c* are unequal, the test should reject H_o or else it makes a type-II error. The difference between the two *c* is the effect size; the larger the effect size, the smaller the number of type-II error should be. We used normal additive noise at a level η of 2.0. Tests were performed with a decision level of 5%. The number of trials *T* was fixed at 400. Each test was replicated a thousand times

Figure A.3 shows the results. When there is 80 blocks ($N = 5$), there are very few type-I errors but the power is very low: The test almost never rejected H_0. In the opposite case (5 blocks with $N = 80$ observations per blocks), the opposite is seen: H_0 is often rejected, resulting in a good power but a type-I error rate near 30%. Choosing the perfect compromise between block size and number of blocks (20 blocks of 20 trials) yielded the best results, with a type-I error rate near 8% and a power near 90% when a large effect size is present. Although the tests were performed with a decision level of 5%, the effective amount of type-I error is slightly larger due to a large amount of covariation within subject (Hoel, 1964).

Insert Figure A.3 about here

In another series of simulations, we tested the efficiency of the test with 1600 trials and, weighting equally type-I errors and power, the test was optimal (equally weighting type-I error and power) at 40 trials per block, suggesting the general rule that the optimal block size for Rao's test is the square root of the number of trials.

Figure Captions

Figure 1. Averaging power curve per block. Top part shows a power curve generated using the parameter $\{a = 0, b = 350, c = 0.455\}$ over 400 trials. Bottom part shows the same curve when averaged by blocks of $N = 40$ trials.

Figure A.1. Bias and efficiency in percentage as a function of the number of trials T for curvature parameter c increasing from top to bottom and noise level η increasing from left to right.

Figure A.2. Bias and efficiency in percentage as a function of block size N for curvature c increasing from top to bottom and noise level η increasing from left to right.

Figure A.3. Percent of times H_0 is rejected using Rao's test with 5% level of confidence as a function of block size N for curvatures of the first simulated data set increasing from top to bottom and curvature of the second data set increasing from left to right. Number of trials T is 400 and noise η is 2.0. The main diagonal contains cases where both curvatures are equal and illustrates the percent of type-I errors. The off-diagonal plots contain cases where curvatures are unequal and thus illustrate the power of the test (one minus the percent of type-II errors). Left part of the first box shows the main diagonal across all c levels. Right part of the first box shows the power across all effect sizes (i.e. for all effect sizes). The second box shows the difference between the power and the type-I errors shown in the first box. Since this scenario weights equally type-I errors and power, the test is optimal at $N = 40$ or $N = 80$. However, if type-I errors are a concern (and weighted more heavily), the test would be optimal at $N = 20$, the square root of the total number of trials.

Table A.1

Overview of the Monte Carlo simulations performed in the Appendix.

Description	**Simulation 1**	**Simulation 2**	**Simulation 3**
Purpose	*Is bias dependant on noise, curvature?*	*Is bias improved by block averages?*	*Is test of linear hypothesis more powerful with averaged data?*
Dependant measures	Bias	Bias	Type-I and Type-II errors
Factors varied	Curvature Sample size Noise	Averaging by blocks Curvature Noise	Curvature of curve 1 Curvature of curve 2 Averaging by blocks
Factors held constant		Sample size (400)	Sample size (400) Noise (2.0)

Notes:

Curvature levels are: 0.2, 0.4, 0.6, 0.8.
Sample sizes T are: 50, 100, 200, 400, 800, 1600.
Noise levels h are: 0.5, 1.0, 2.0.
Block sizes N are: 1 (no block average), 5, 10, 20, 40, 80

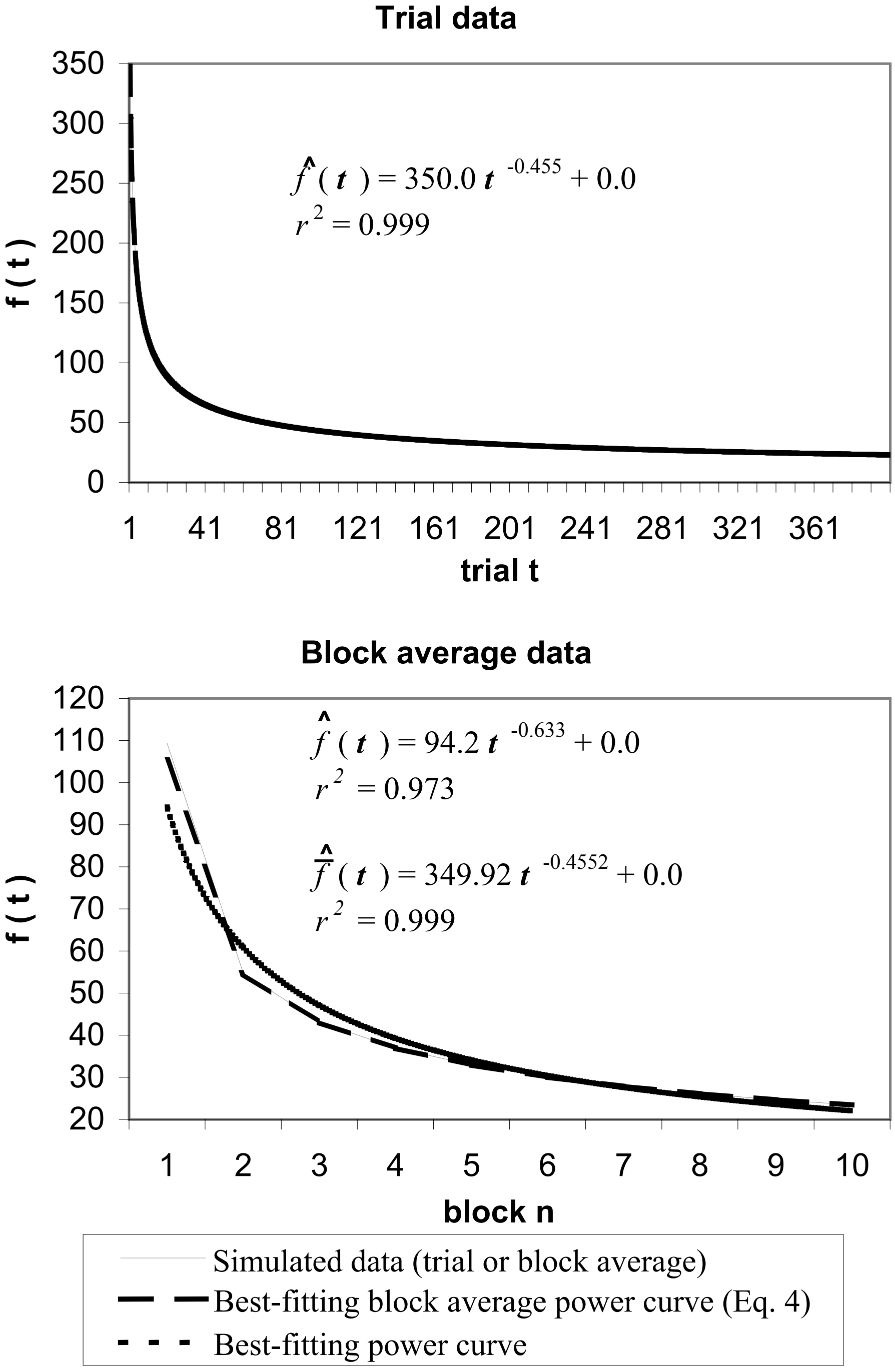
Trial data
350
300
250
200
150
100
50
0
f (t)
1
41
81
121
161
201
241
281
321
361
trial t
$\hat{f}(t) = 350.0\, t^{-0.455} + 0.0$
$r^2 = 0.999$
Block average data
120
110
100
90
80
70
60
50
40
30
20
$\bar{f}(t)$
1
2
3
4
5
6
7
8
9
10
block n
$\hat{f}(t) = 94.2\, t^{-0.633} + 0.0$
$r^2 = 0.973$
$\hat{\bar{f}}(t) = 349.92\, t^{-0.4552} + 0.0$
$r^2 = 0.999$
Simulated data (trial or block average)
Best-fitting block average power curve (Eq. 4)
Best-fitting power curve

Figure 1

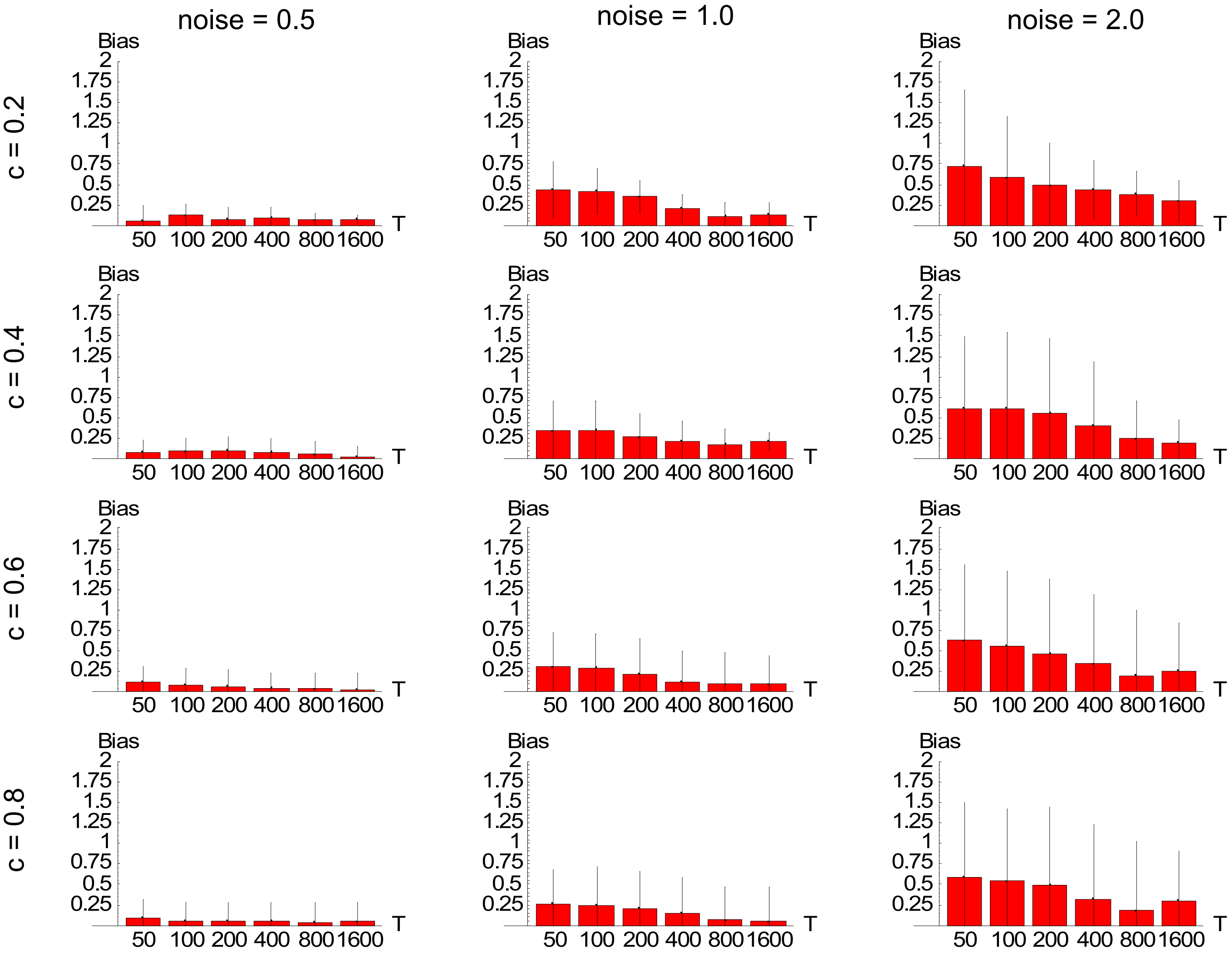
noise = 0.5
noise = 1.0
noise = 2.0
c = 0.2
c = 0.4
c = 0.6
c = 0.8
Bias
2
1.75
1.5
1.25
1
0.75
0.5
0.25
T
50
100
200
400
800
1600

Figure A.1

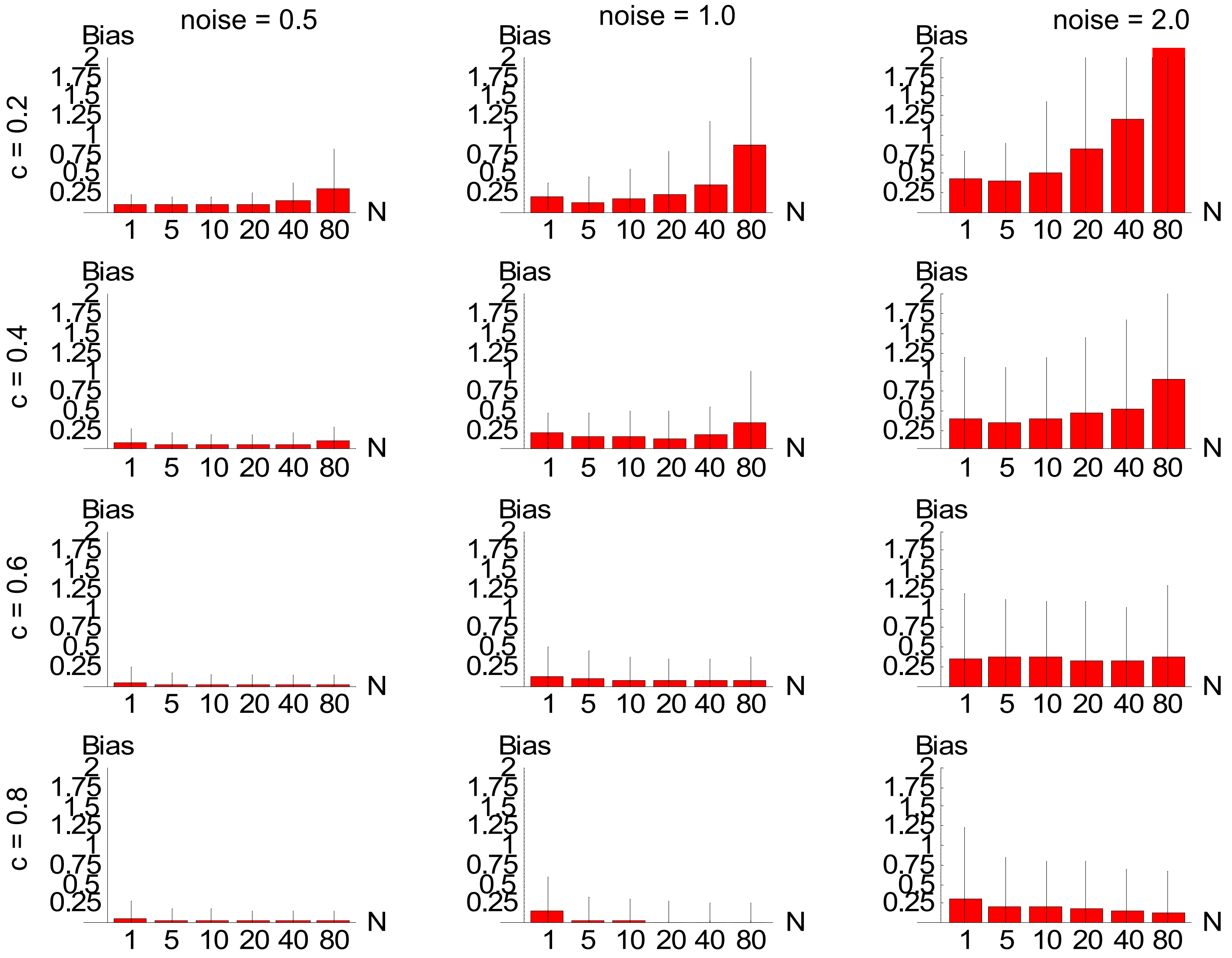
noise = 0.5
noise = 1.0
noise = 2.0
c = 0.2
c = 0.4
c = 0.6
c = 0.8
Bias
2
1.75
1.5
1.25
1
0.75
0.5
0.25
N
1
5
10
20
40
80

Figure A.2

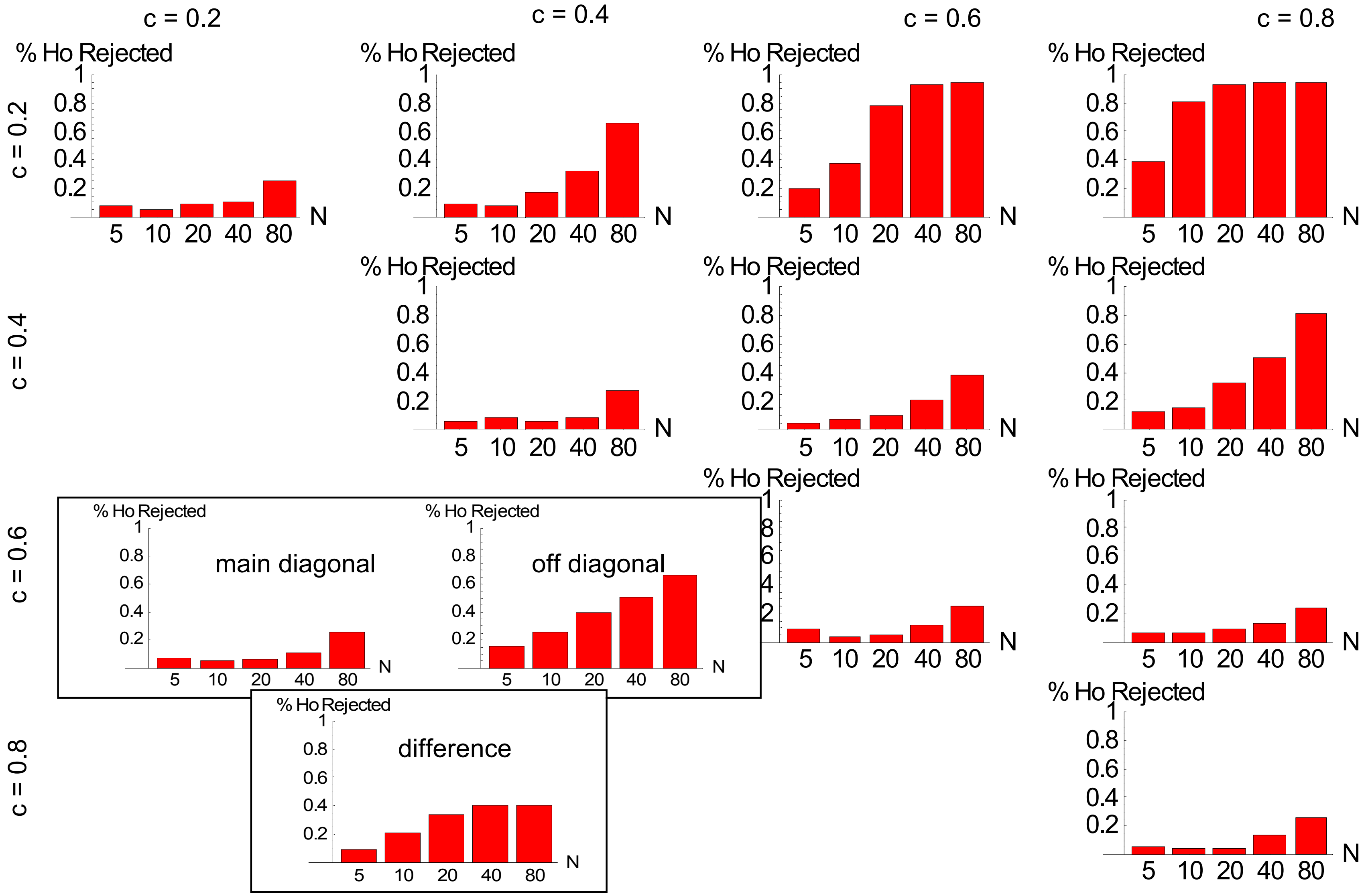
c = 0.2
c = 0.4
c = 0.6
c = 0.8
c = 0.2
c = 0.4
c = 0.6
c = 0.8
% Ho Rejected
1
0.8
0.6
0.4
0.2
N
5
10
20
40
80
main diagonal
off diagonal
difference

Figure A.3

Listing 1. A Mathematica program that performs a test of linear hypothesis (Rao, 1959). It reads the input file "data.dat" which is composed of s columns with T observations in each. Comments are enclosed between (and *).*

```
(******* load a useful package and set working directory *******)
Needs["Statistics`MultiDescriptiveStatistics`"]
SetDirectory["C:\\WINDOWS\\Bureau\\"];

(********************** model information *********************)
Model[t_, c_] := {1, t^-c}      (*trial data power curve*)
θ[a_, b_] := {a, b}
s := 4                          (* number of columns*)

(* definition of the Sum of Square Error used for minimization *)
                            T
SSE[set_, a_, b_, c_] :=    ∑ (set⟦t⟧ - θ[a, b].Model[t, c])²
                           t=1

(********************* read the data file *********************)
FileFormat = Table[Real, {s}];
data = ReadList["data.dat", FileFormat];
T = Length[data]

(****************** compute the summary values *****************)
y = Mean[Transpose[data]];
S = CovarianceMatrix[Transpose[data]];

(******* performs a fit over the average data and keep c *******)
GroupFit = FindMinimum[SSE[y, a, b, c],
   {a, 100, 300}, {b, 400, 2000}, {c, 1.2, 2.0}
  ]⟦2⟧
c = c /. GroupFit

(***** performs a fit for each column and average a and b ******)
IndividualFit = Table[FindMinimum[SSE[Transpose[data]⟦i⟧, a, b, c],
     {a, 100, 300}, {b, 400, 2000}, {c, 0.2, 1.0}
    ]⟦2⟧,
   {i, 1, s}
  ];
θ̂ = Mean[θ[a, b] /. IndividualFit]

(******************* instantiate the model *******************)
A = Table[Model[t, c], {t, 1, T}];

(********** Perform Rao' s test of linear hypothesis ***********)
r = Length[θ[a, b]] + 1;
n = s ;
     n - r
F =  ----- (y - A.θ̂).PseudoInverse[S].(y - A.θ̂)
       r
```