Getting parameters from learning data

Denis Cousineau
Université de Montréal

Guy L. Lacroix
Carleton University

Response time data in learning experiments show a typical trend. They start out slow, quickly improve, before finally tending toward optimal performance. This trend provides critical information that can be used to test various theories of learning. One convenient way to characterize the data is the use of a learning curve; an idealized curve that passes through the observed data points as a function of training. This idealized curve has free parameters that must be estimated using optimization techniques. In this tutorial, we show how to estimate learning curve parameters using three softwares (Excel, SPSS, and Mathematica) assuming that the idealized curve is a power function. The techniques can easily be adapted to other functions. Finally, details are provided on related topics (maximizing block sizes, testing curvatures, etc.).

With practice comes improvement. When people are learning to accomplish any kind of task, whether it is swimming, riding a bike, using a keyboard, or solving Sudoku problems, it is well-known that they start out slowly, but that they quickly become better. Ultimately, it is only when people become very good at doing a task that they need a lot of practice to see noticeable improvements. Intuitively, this process is easy to understand. To revisit the swimming example, it only takes a few lessons to learn to swim, but it takes years of practice to meet the swimming times necessary to compete at the Olympics.

Many theories seek to explain how performances evolve with training (for instance, Anderson, 1992; Logan; 1988; Palmeri & Nosofsky, 1997). This is a challenging endeavor because it is difficult to quantify performances times across training (which I will now call the longitudinal results) accurately. The most convenient method to quantify longitudinal results is to create an idealized curve, called the learning curve. This learning curve may be described by a mathematical function and it relates training to unknown learning variables called parameters.

Once estimated, learning curve parameters can be used to test learning theories. For example, it can be shown that the Exemplar-Based Random Walk (EBRW) model (Palmeri & Nosofsky, 1997) predicts only one rate of learning (Cousineau, Lacroix, Giguère & Hélie, in preparation). So, if a parameter that captures the rate of learning is shown to vary systematically according to some experimental manipulation, then EBRW would have to be modified. Another example is Logan's (1988) Instance-Based Theory of Automaticity. It predicts that the learning rate should be the same whether it is measured by mean RTs or by standard deviations of RTs. Thus, if one were to show that this prediction is incorrect, then one would conclude that Logan's theory is problematic. Finally, transfer theories (such as Fisk & Schneider, 1983) suggest that learning rates before and after transfer are unrelated. All these predictions provide strong tests of the models, and these tests would be impossible if only one part of the learning data, such as the asymptotic results, were considered.

In this tutorial, we present the learning curve being celebrated in this special issue, Newell and Rosenbloom's (1981) power curve, which they called "The power law of practice". We then explain how to estimate the parameters of the power curve using *Excel*, *SPSS* and *Mathematica*. Finally we end the tutorial with various details pertaining to statistical tests.

The power curve

The power curve has three parameters that quantify the learning data: the asymptote (the stabilized performance at the end of training), the amplitude of the initial block (the difference between initial performance and the asymptote), and the curvature (learning rate). Whereas the asymptote and the amplitude are very intuitive and can be estimated from a visual inspection of the data, the curvature parameter is abstract and requires an underlying mathematical curve to be quantified.

Newell and Rosenbloom's (1981) power law of learning is a power curve given by

$$P(N \mid a,b,c) = a + bN^{-c} \qquad (1)$$

in which the predicted performance P (often RT but also percent correct) at block N is a function of three unknown parameters a, b and c. The parameter a is the asymptote parameter, b is the amplitude parameter, and c is the curvature parameter.

Assuming this curve, initial performance (at block $N = 1$) is given by $a + b$, that is, the asymptote augmented by the amplitude. Final performance (when $N \rightarrow \infty$) is given by a. Heathcote et al. (2000) noted that the decrease in RTs with training follows a square law, so that if N blocks are required to reduce performance by 50%, N^2 blocks are required to reduce performance by another 50% (hence, to reduce performance to 50% × 50% = 25%). Figure 1 shows a power curve with its parameters.

Before closing this section, we would like to make a theoretical note. Although it was dubbed the "power law of learning", its status as a law is contentious. The power curve is a convenient tool to quantify learning results and it seems to explain learning data well. Nevertheless, we believe that its status as THE law of learning is not firmly established for at least two reasons: i) the logical necessity of this mathematical curve has not yet been demonstrated outside of specific models of learning (although it is within some
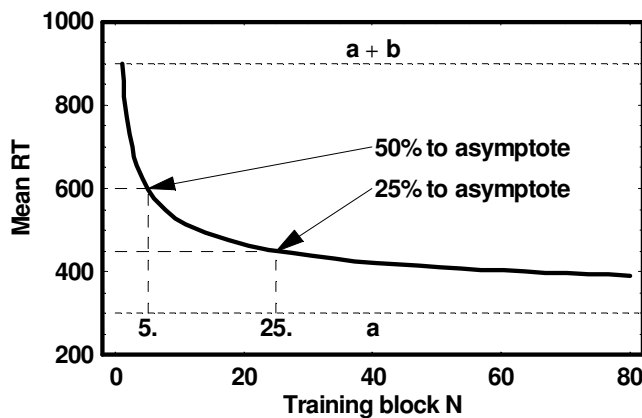

Figure 1. An example of a power curve with parameters {a = 300, b = 600, c = 0.43}. Performance is reduced by 50% at block 5 and by another 50% at block 25 (which is 5². This relation holds for any percentage.

specific models, such as ACT*, Anderson, 1992 and IBTA, Logan, 1988); ii) the data available from highly practiced participants are limited because the attrition in long-lasting learning studies is quite high. Typically, participants drop out of these experiment at a rate of 50% every 10 sessions (Czerwinski, Lightfoot, & Shiffrin, 1992, Cousineau & Shiffrin, 2004). If more of these data were available, it would allow for more powerful tests of the Power Law as well as other curves such as the exponential (see Heathcote, Brown & Mewhort, 2000) and other alternatives (Rickard, 1997; Stevens & Savin, 1962; Cousineau, Goodman, & Shiffrin, 2002).

Learning curve fitting

In order to estimate learning curve parameters, three ingredients are necessary. First, one needs a mathematical curve that can capture the desired characteristics of the learning data. This curve has a certain number of unknown learning variables called free parameters. Second, one requires an objective function describing the adequacy of the fit between the mathematical curve and the actual data. Finally, one must use an optimization procedure that explores the possible parameter values until the best fit is found.

In the previous section, we presented the power curve, which is the first ingredient needed. The second ingredient is the function that determines the fit between the learning data and the mathematical curve. The fit is generally computed using the sum of squared error (*SSE*), but it may also be computed using other measures such as the root mean square error (*RMSE*). Note that the choice of measure does not change the value of the parameters obtained.

The *SSE* computes the sum of the squared error between the learning curve and the data for each block and is formally computed using:

$$SSE(a,b,c) = \sum_{i=1}^{n} \left(p_i - P(i \mid a,b,c) \right)^2 \qquad (2)$$

in which n is the number of blocks, p_i is the observed data at block i, and $P(i \mid a, b, c)$ is the predicted performance given by the mathematical curve (e.g. Eq. 1). The fit depends on the choice of the free parameters {a, b, c}, which, in turn, alter the shape of the predicted curve P. This objective function is a sum over the blocks 1 through N. If there are blocks for which the performance is unavailable, the summation is over blocks where data are available.

Finally, the optimization procedure can either be a gradient descent, a quasi-newton search, or the simplex. The simplex method is easily available and yields very good results (see Cousineau, Brown & Heathcote, 2004, for a quick review). Although these procedures are very complex, there is no need to program them as they are

	A	B	C	D	E	F	G	H	I
1	**Fitting learning curves**								
2									
3									
4	Put first-guess parameters on the right,				Parameters				
5	then call the solver to minimize the SSE.				asymptote		471		
6					amplitude		3360		
7					curvature		1.09		
8									
9									
10			**Data**				**Model**		
11	*Block*		*Mean RT*		*Predicted*		*Squared error*		*SSE*
12	1		3765		3830		4213		1071021
13	2		2076		2044		1006		
14	3		1800		1480		102157		
15	4		1353		1208		21158		
16	5		931		1048		13738		
17	6		862		944		6693		
18	7		902		870		1013		
19	8		890		816		5538		
20	9		643		774		17053		
21	10		556		741		34264		

Figure 2. A portion of an Excel spreadsheet that computes the power curve predicted performance and the SSE given some data (first two columns) and first-guess parameter values (top right of the spreadsheet). These first-guess values will be replaced by the best-fitting values after the optimization procedure (called Solver in Excel) is called.

spreadsheet is available on the journal's web site. The predicted performance is a function of the block number (available in the first column) and of the parameter values to be estimated. First-guess values of the parameters are indicated in the top of the spreadsheet: $\{a = 400, b = 3000, c = 1.0\}$. The first guess for a and b are obtained through a visual inspection of the data. The parameter c was set to 1.0 because the value of c obtained in this type of experiment is usually between 0.2 and 1.5. The predicted performance (e.g. cell E12) is obtained using the *Excel* formula:

= I5 + I6 * A12 ^ -I7

in the case of the first predicted performance (A12 contains the block 1). The squared error column (i. e. column G) is computed using the formula:

= (E12 - C12) ^ 2.

Finally, the *SSE* (Eq. 2) is the sum of all the squared error over the 60 blocks, in this case:

= SUM (G12 : G71)

included in many software products, including *Excel*, *Mathematica*, *Matlab*, *SPSS*, and *S*.

Note that constraints can be given to the parameters. For example, one can constraint the asymptote to be at least 250 ms so that it has psychological plausibility (more on that topic later). Furthermore, the parameter c can only take a positive value and a constraint must be given to achieve this end (otherwise the curve has no asymptote). Regarding the parameter b, it must be positive in the case of RT learning data so that the initial performance is *above* the asymptotic performance. However, for accuracy data, the parameter b must be constrained to be negative, such that initial performance is *below* the asymptotic level of accuracy.

An example of fitting the learning curve to data using Excel

The following briefly shows how to obtain best-fitting parameters using the power curve of Eq. 1. and data from a single participant in a visual search experiment (Cousineau, in preparation). There are 60 blocks of 60 trials, collected over 10 successive sessions. The dependant variable is the mean response time for the correct responses.

Figure 2 shows part of an *Excel* spreadsheet in which the available data are in column C, and the predicted performance given by Eq. 1 are in column E. The complete

To find the best-fitting parameters (that will replace the initial guesses), call the *Solver* in the *Tools* menu (if not installed, go to *Add-ins* in the *Tools* menu) (see the screen capture of Figure 3). The target cell is the objective function to be minimized, that is, the cell containing the *SSE*. Check the radio button *Min* to indicate that the objective function must be as small as possible (not the reverse). The variable cells are the ones containing the parameter values that can be explored in order to find the best fit. Finally, the constraints are used to specify the allowable range of values for the parameters.

Once the *Solve* button is pressed, the search begins and the resulting best-fitting parameters replace the first-guess parameters. *Excel* finds $\{a = 471, b = 3360, c = 1.09\}$. The resulting fit is 1,070,690 ms², or equivalently, a *RMSE* of 133 ms ($\sqrt{SSE / n}$)). This number indicates the typical (unsigned) error between one point and the curve (measured vertically). Figure 4 shows the data along with the best-fitting power curve.

The same example of fitting the learning curve to data using SPSS

Although *Excel* works well, it may be cumbersome to use when many participants must be evaluated, because the data must be manually split into individual data, and the
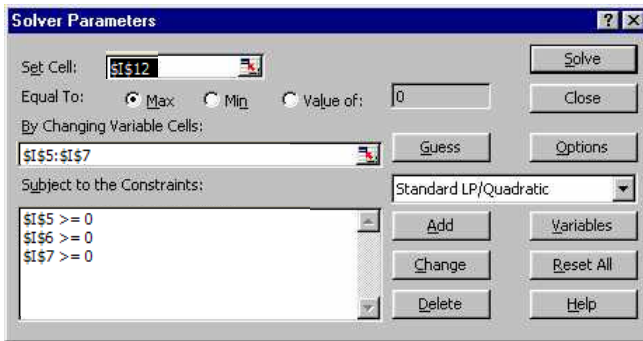
Figure 3. The dialog that appears when the Solver is called.

Solver called for each. One solution is to use programmable software that can split data files or extract only portions of the data.

SPSS (versions 11 and up) has a command (*CNLR*) that performs minimization using the *SSE* objective function (called the loss function in *SPSS*). The minimum specification in the command *CNLR* is the name of the column containing the observed data and the name of the column containing the predicted performance. The *SSE* function is not indicated, as it is implicitly assumed. Constraints can be given with the option /*bounds*.

A short SPSS syntax program that analyses the data of the previous example is provided in Listing 1. It assumes that the data are in a tab-separated text file called "data.txt", containing two columns, *blocks* and *rt*. The predicted performance will be computed in a column called *pred*.

Once executed, the results are displayed in the *Output* window. Note that the best-fitting parameters and the fit are absolutely identical to those found using *Excel*. *SPSS* also provides approximate confidence intervals for the parameters and the possible correlations between the parameters in the model. Such information cannot be obtained using *Excel*.

An example of fitting the learning curve to data using Mathematica

In the following, we show the same example using *Mathematica*, assuming that the data are in a text file called "data.txt" (see Listing 2).

The minimization is done using the command *NMinimize,* which specifies (on the second line) the objective function and the constraints, and (on the third line) the parameters with an approximate range in which to start searching.

The line labeled "Out[5]=" by *Mathematica* shows the results. The best fit value is first, followed by the parameter values to obtain such the fit. Once again, the solution found is identical to those generated by *Excel* and *SPSS*.

Various considerations

I finish this brief tutorial with various considerations that pertain to questions that are frequently asked about learning curve fitting. First and foremost, learning curve data should be fit for each individual participant. Indeed, the average curve for a group of participants is not necessarily a power curve even if all the individual curves are power curves (Estes, 1956, Cousineau, Hélie & Lefebvre, 2003). This is also true when one assumes an exponential curve (Heathcote, Brown & Mewhort, 2000) or any other type of curve.

Reasonable estimates for parameters *a* and *b* can be obtained using Min(p_i) and Max(p_i) – Min(p_i) respectively. That way, "non-parametric" estimates can be obtained for the asymptote and the amplitude. By "non-parametric", we mean estimates that are not based on a specific mathematical curve, but that would be roughly correct independently of the type of curve assumed. Note that such a "non-parametric" estimate for the curvature presently exists. Because these non parametric estimates are not based on a mathematical model, they may be seen as less controversial (by reviewers, for example).

Whenever possible, you should maximize the number of blocks, as well as the number of observations within a block. Indeed, if one splits the data in 2 blocks, then there would not be enough data points to accurately fit a power curve. Inversely, if one generated blocks containing only two observations, then the data would be very noisy and again, the fit would be unreliable. The best compromise is to split the *t* trials into $\sqrt{t}$ blocks of $\sqrt{t}$ trials (Cousineau et al., 2003). For example, if you have a total of 400 observations, split them into 20 blocks of 20 observations. Those blocks may be created *post hoc* and do not have to correspond to actual blocks of training as experienced by the participants. However, one must keep in mind that the learning rate estimate changes whenever the number of trials per block is changed. Hence, to compare *c*s across experiments, one must make sure the blocks are of equal size (or see Cousineau, Hélie & Lefebvre, 2003, to obtain *c* estimates that do not depend on block sizes).

Statistical tests

The parameters can be used to test some predictions. For example, two training techniques could be compared, in which case the null hypothesis would be that the asymptotes do not differ. Another example could be to see whether high performers learn faster than a control group, in which case the null hypothesis would be that the learning rates (curvatures) are equal between the two groups. When one considers using statistical tests, the rule is that whenever a test on the original data is possible, this test should be preferred over a test on the parameters. Indeed, the
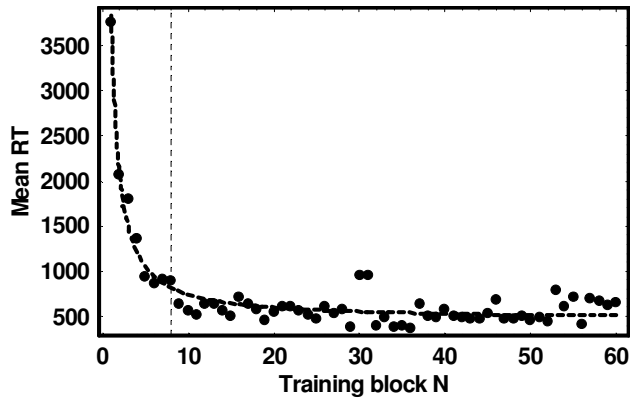
Figure 4. The mean response times from the data analyzed in the previous figure. The dashed line shows the moment where the data are no longer showing any significant improvement (*p* > .10).

optimization procedure is never guaranteed to find the true parameters (the best-fitting parameters may be different from the true parameters, Hélie, 2006). As such, this extra step introduces further variance, which may reduce statistical power. A statistical test to decide if a group of participants has equal curvatures is described in Cousineau et al. (2003). This test requires an estimated curvature for the group data. However, it requires many participants (in the hundreds), which is considerable considering that each participant must be trained for many consecutive days. A test to decide if two groups have the same learning rate can be done on the estimated curvature parameter c (e.g. a simple t test if there are only two groups). To test if two groups (or more) have reached the same asymptote, prefer a test on the performance on the last block rather than a test on the estimated asymptote parameters, provided that a plateau is visible in the data. Finally, a test of the amplitude can be replaced by a test on the first block of data to which the smallest data are removed. Note, once again, that these operations must be conducted on individual data.

Has the asymptote been reached?

To decide if a participant has reached asymptotic performance, one can use the following procedure. Note that formally speaking, no one ever reaches asymptotic performance. This is because it is postulated by most learning theories that an infinite amount of practice is necessary to reach it. Hence, this question is replaced by the following one: is there still significant improvement in the performance after a given number of blocks? To answer this question, one can run a regression on performance as a function of block number over all blocks of data. If the regression slope is significantly different from zero (and negative), the first block may be removed and the procedure may be repeated. When one reaches a point where the regression slope is no longer significantly different from

zero, one can conclude that asymptotic performance has been reached. Considering the large number of regression slopes tested, one should use a less severe decision criterion (such as $p > .10$). [1]

Regarding the data analyzed in the previous section, the asymptote seems to be reached very early, as the regression slope does not differ from zero past the 8th block ($r = -.18$, $t = 1.21$, $p = .12$). The vertical dashed line in Figure 4 indicates the point at which asymptote was reached. Such a fast stabilization of performance is rather uncommon because, in other experiments (e.g. a same-different paradigm), it often takes 10 sessions before a significant plateau is reached.

Low-asymptote pathology

Heathcote et al. (2000) raised another concern regarding power curve fitting. They noticed that when a power curve was fit to empirical data, the best-fitting asymptote was often close to zero (especially in Logan, 1988). Although such a solution provides the best fit, it is unrealistic from a psychological point of view and should be rejected. This problem is inherent to the nature of the power curve. Indeed, as noted above, the rate of descent becomes smaller as N increases. Hence, there comes a point where performance is almost flat even though it may still be far from the asymptote. Because of this, the fitting procedure has no clear cue as to where to put a: Based on fitting only, the asymptote can be anywhere below the lowest performance. To avoid what Heathcote et al. have called the "low-asymptote pathology", a few remedies have been proposed. The least controversial is to start the search with a first-guess asymptote close to the observed minimum. The minimization procedure tends to stay close to first-guess values and so is less likely to return zero for the best-fitting asymptote. In *Excel*, it would mean using as a first guess for a the value $Min(p_i)$. Although this simple heuristic can eliminate a large number of pathological fits, there are data sets for which the solution found will still be unrealistic. A second possibility (which can be used in conjunction with the first) is to have a more realistic constraint on a. Instead of having $a > 0$, we might arbitrarily use $a > 250$. Now this solution is not very satisfactory because the quantity 250 ms is not theoretically justified, and again, a certain proportion of fit will indicate that the best-fitting asymptote is 250 ms. A final possibility is to assume that the best-fitting asymptote should be close to the best performance observed for each participant. This implicitly assumes that the participant ended the experiment with quasi-optimal performance. Hence, if there is no indication of an apparent asymptote in the data, do not use the following. To impose a solution in which the asymptote is reasonably close to the best performance, a penalty term can be added to the objective function. For example,

$$SSE(a,b,c) = \sum_{i=1}^{n} \left(p_i - P(i\,|\,a,b,c)\right)^2 + \left(Min(p_i) - a\right)^2 \quad (3)$$

adds a quadratic penalty to the *SSE*. The greater the distance between the proposed asymptote parameter *a* and the best performance, the greater the penalty will be.

When used with the data of the above participant, the results do not change much because the original solution was not affected by the low-asymptote pathology. We get the following best-fitting parameters {*a* = 466, *b* = 3360, *c* = 1.09}.

Taking into consideration both mean RT and percent correct

When studying response times in tasks for which a correct response must be given, there may be blocks where the participant, for various reasons including fatigue, may decide to trade accuracy for speed. Thus, there would be points on the learning curve where mean RTs are low, but where accuracy (percent correct) is also low. When this occurs more than once, it gives the learning curve a ragged look. In such conditions, finding the best-fitting parameters can be more difficult.

Townsend and Ashby (1983) proposed this elegant solution, which consists in considering mean RTs after they have been weighted by accuracy. Hence, when accuracy is close to 100%, the mean RT is unchanged, but when accuracy is low, mean RT is increased to compensate for the trade-off between speed and accuracy. This is achieved using the following transformation:

$$p_i = \frac{\overline{RT_i}}{\overline{ACC_i}} \qquad 4$$

where $\overline{RT_i}$ is the mean RT at block *i* and $\overline{ACC_i}$ is the accuracy at the same block. The resulting learning curve no longer represents response times as a function of blocks, but rather response time per percentage of accuracy as a function of blocks, and consequently, the best-fitting parameters on the transformed data may be very different from those found on the RT data only.

Discussion

In this tutorial, we showed how to get best-fitting parameters from learning data. This kind of technique is very easy to execute because optimization procedures are now widely available (this was not the case when I published Cousineau & Larochelle, 1997). Moreover, the whole process can be automatized using software such as *Mathematica* or *Matlab*. Objective functions such as the *SSE* (and related measures) should be used when the distribution of the raw data is unknown (as is the case with response times but not for accuracy; see Cousineau, Charbonneau & Jolicoeur, 2006). When the distribution of the raw data is known, it is always preferable to use an objective function based on the likelihood function. Indeed,

this function is generally the most efficient method to find best-fitting parameters (Cousineau, Brown & Heathcote, 2004).

Information that can be estimated from an examination of learning curves is currently undervalued. It is our impression that stronger and more varied tests of learning theories could be obtained if such information were considered more rigorously. We hope that this quick tutorial will encourage researchers to give more attention to learning curves. The reader will find on the journal's web site an *Excel* spreadsheet that computes the *SSE* along with a second spreadsheet that finds where the asymptote is reached. One can also find a short *Mathematica* program that finds the best-fitting parameters using *SSE* (Listing 2) but also using the *RMSE* and using the penalty against the low-asymptote pathology along with a tab-separated text file containing the data.

References

Anderson, J. R. (1992). Automaticity and the ACT* theory. *American Journal of Psychology*, 105, 165-180.

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

Cousineau, D., Brown, S., & Heathcote, A. (2004). Fitting distributions using maximum likelihood: Methods and packages. *Behavior Research Methods, Instruments, & Computers*, 36, 742-756.

Cousineau, D., Charbonneau, D., & Jolicoeur, P. (2006). Parameterizing the attentional blink effect. *Canadian Journal of Experimental Psychology*, 60, 175-189.

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

Cousineau, D., Hélie, S., Lefebvre, C. (2003). Testing curvatures of learning function on individual trial and block average data. *Behavior Research Methods, Instruments, & Computers*, 35, 493-503.

Cousineau, D., & Shiffrin, R. M. (2004). Termination of a visual search with large display size effect. *Spatial Vision*, 17, 327-352.

Czerwinski, M., Lightfoot, N. & Shiffrin, R. M. (1992). Automatization and training in visual search. *American Journal of Psychology*, 105, 271-315.

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

Fisk, A. D. & Schneider, W. (1983). Category and word search: generalizing search principles. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 9, 177-195.

```
COMMENT ******************* STEP 1: READING THE DATA *************************************.
COMMENT change the following path according to the location of you data file.
cd "C:\Documents and Settings\cousined\Bureau\Terminées \32-CurveFitting\demoFiles".

COMMENT open the data, in a tab-separated text file named "data.txt".
GET DATA  /TYPE = TXT
 /FILE = 'data.txt'
 /DELIMITERS = "\t"
 /VARIABLES = block F2.1 rt F4.2.

COMMENT ******************* STEP 2: DEFINING THE MODEL **********************************.
COMMENT create temporary variable with starting values.
MODEL PROGRAM a=300 b=500 c=2 .
COMMENT define the predicted performance.
COMPUTE pred = a+b * block ** -c.

COMMENT ******************* STEP 3: CALLING THE MINIMIZATION ALGO *******************.
CNLR rt
 /PRED pred
 /BOUNDS a >= 0; b >= 0; c >= 0.
```

Listing 1. A SPSS syntax file that read a tab-separated text file containing two columns called block and rt. It then creates temporary variables a, b and c with first-guess values. Finally, CNLR (Constrained Non-Linear Regression) run a search for the optimal parameter values.

Step 1- Reading the data

Assumes the following format: one column for block number,the second for mean response time.

```
mypath = "C:\\Documents and Settings\\cousined\\Bureau\\CurveFitting\\";
mydata = Import[mypath <> "data.txt", "TSV"];
```

Step 2- Defining the model

$$PC[N\_, \{a\_, b\_, c\_\}] := a + b \, N^c$$

$$SSE[data\_, \{a\_, b\_, c\_\}] := \sum_{i=1}^{Length[data]} (data[[i, 2]] - PC[data[[i, 1]], \{a, b, c\}])^2$$

Step 3- Calliing the minimization procedure with constraints

```
sol = NMinimize[
  {SSE[mydata, {a, b, c}], a > 0 && b > 0 && c > 0},
  {{a, 100, 500}, {b, 500, 1000}, {c, 0.1, 0.5}}
 ]
```

$\{1.07069 \times 10^6, \{a \to 470.694, b \to 3359.28, c \to 1.09423\}\}$

Listing 2. A short Mathematica program that reads a tab-separated (format TSV) text file containing two columns: the block number and the corresponding mean response time. It then defines (:=)what are a power curve (PC) and SSE. Finally, it calls the optimization procedure with the command NMinimize. The sign data[i,j]represents the data in the ith line, jth column.

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

Hélie, S. (2006). An introduction to model selections: Tools and algorithms. *Tutorials in Quantitative Methods for Psychology*, 2, 1-10.

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.

Nosofsky, R. M. & Palmeri, T. J. (1997). An Exemplar-Based Random Walk Model of Speeded Classification. *Psychological Review*, 104, 266-300.

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.

Stevens, J. C. & Savin, H. B. (1962). On the form of learning curves. *Journal of Experimental Analysis of Behavior*, 5, 15-18.

Townsend, J. T. & Ashby, F. G. (1983). *Stochastic Modeling of Elementary Psychological Processes*. Cambridge, England: Cambridge University Press.

[1] This will inflate the type-I error rate (obtaining a slope significantly different from zero). Because we are looking for exactly the opposite, it results in a very conservative test of the null effect (slope not significantly different from zero).