Quantitative Methods for Cognitive Scientists

Basic Parameter Estimation

Emre Neftci

Department of Cognitive Sciences, UC Irvine,

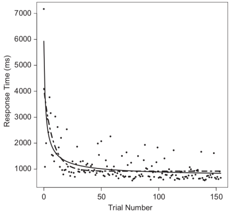
April 18, 2019

Emre neftci
eneftci@uci.edu

https://canvas.eee.uci.edu/courses/16991

Example Model in Cognitive Science: Skill acquisition

Participants judged the numerosity of random patterns having between 6 and 11 dots.



Heathcote et al. 2010

Two different functions fit the data well:

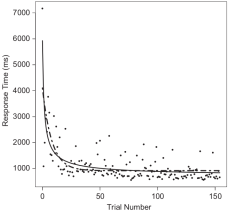
- Exponential (dashed): $RT = e^{-\alpha N}$
- Power (solid): $RT = N^{-\beta}$

where N is the number of trials, RT is reaction time and α, β are parameters.

Example Model in Cognitive Science: Skill acquisition

The benefits from practice follow a nonlinear function: Improvement is rapid at first but decreases as the practitioner becomes more skilled

Thorndike, 1913



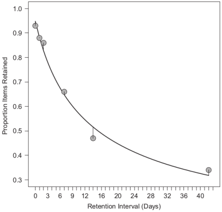
Two different functions fit the data well:

- Exponential (dashed): $RT = e^{-\alpha N}$
- Power (solid): $RT = N^{-\beta}$

where N is the number of trials, RT is reaction time and α, β are parameters.

Example Model: Memory recall example

Participants studied a set of 60 obscure facts (e.g., "greyhounds have the best eyesight of any dog"), and their memory for those facts was tested after 5 minutes, and again 1, 2, 7, 14, 42 days later.



Carpenter et al., 2008

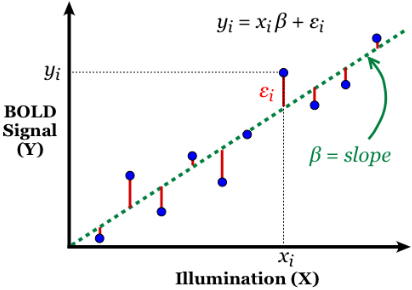
For example: Predicted probability of recall, as a function of time:

$$p(t) = a(bt + 1)^{-c}$$

a, b, and c are the three parameters of the function. Measure data is the number of correct responses.

Example Model: OLD signal in fMRI

“Blood oxygen level dependent (BOLD) signal in functional magnetic resonance imaging (fMRI) data assume that the BOLD signal is predominantly linear in space and time.”



$$y_i = x_i\beta + \epsilon_i$$

- y is the BOLD signal (what is measured with fMRI)
- x is a value determined by the experiment design (retinal illumination here)

Hansen et al. 2004, Parametric reverse correlation reveals spatial linearity of

The goal of parameter estimation is to find those parameter values that maximize the agreement between the model's predictions and the data.
The extent of that agreement then tells us something about the utility of the model.
Examples:

- Fit α and β in the exponential vs. power model for skill learning
- Estimate parameters, μ and σ in the sequential sampling model.
- Fit α and β in the fMRI example
- Fit a, b and c in the memory recall example
- Later in this class: train a neural network

Further reading, Chapter 3, Farrell and Lewandowsky,, 2018

A parameter estimation problem generally has three components:

1. **A model or function**: the free parameters of this functions are estimated
2. **A cost function**: Depending on the domain, also called loss, discrepancy, utility, reward etc. Cost functions are generally scalar (when evaluated they return a single number)
3. **An optimizer**: an engine that searches for the best parameters given the data.

Cost function in continuous cases

Most parameter estimation procedures try to minimize the discrepancy between predictions and data.

- If the data and predictions are continuous (= any real number), then a common cost function is the Root Mean Square Error (RMSE):

  $$RMSE = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(d_n - y_n)^2}$$

  N is the number of data points, $d_1, ..., d_N$ represents data and $y_1, ..., y_N$ represents prediction.
- This approach is sometimes called "least squares"

Example: calculating an RMSE

Two data samples (N=2)
Data: $d_1 = .5, d_2 = .6$
Assume our model predicts: $y_1 = .3, y_2 = 2.2$
Plugging these values in the RMSE, we get:

$$RMSE = \sqrt{\frac{1}{2}\left((.5 - .3)^2 + (.6 - 2.2)^2\right)} = 1.613$$

Example: calculating an RMSE

Two data samples (N=2)
Data: $d_1 = .5, d_2 = .6$
Assume our model predicts: $y_1 = .3, y_2 = 2.2$
Plugging these values in the RMSE, we get:

$$RMSE = \sqrt{\frac{1}{2}\left((.5 - .3)^2 + (.6 - 2.2)^2\right)} = 1.613$$

- Now, find the values of y_1 and y_2 that would minimize the RMSE

Example: calculating an RMSE

Two data samples (N=2)
Data: $d_1 = .5, d_2 = .6$
Assume our model predicts: $y_1 = .3, y_2 = 2.2$
Plugging these values in the RMSE, we get:

$$RMSE = \sqrt{\frac{1}{2}\left((.5 - .3)^2 + (.6 - 2.2)^2\right)} = 1.613$$

- Now, find the values of y_1 and y_2 that would minimize the RMSE
- Generally y is a function and we fit the parameters of this function

- For example, assume the function:

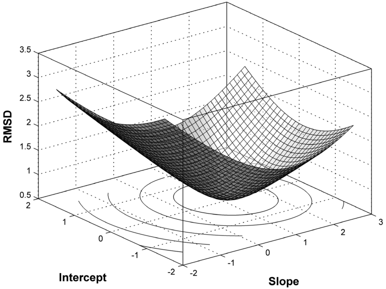
  $$y_n = \alpha + \beta x_n$$

  where ϵ_n here is an error term ($\epsilon_n \sim N(0,1)$).
- The goal is to find the parameters α (intercept), and β (slope), such that the RMSE is minimized.

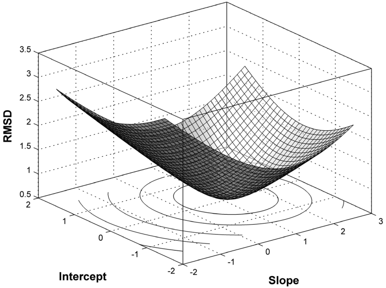
The Cost Surface

An example RMSE cost surface for

$$y_n = \alpha + \beta x_n$$



- The point where the RMSE is minimized
- One or more minima will *always* exist
- The existence of a minimum does not guarantee that the model can adequately fit the data to which it is applied



Parameters can be fit in many ways:

- Visual: find visually the minimum of the cost function by plotting it (e.g. as above)
- Grid Search: examine all possible combinations of parameter values and pick the best fitting one
- Iteratively: Take small steps in the direction of decreasing RMSE ("downhill"). There are many algorithms and approaches for doing this.

Visual and grid search do not work when the number of parameters is high. We focus in this class on iterative methods

Cost function in discrete (categorical) cases

When the number of responses is constant but each response can fall into one of several different categories, one can use either one of the two functions (χ^2 or G^2)

- Chi-squared error function:

$$\chi^2 = \sum_{j=1}^{J} \frac{(d_j - Np_j)^2}{Np_j}$$

Cost function in discrete (categorical) cases

When the number of responses is constant but each response can fall into one of several different categories, one can use either one of the two functions (χ^2 or G^2)

- Chi-squared error function:

$$\chi^2 = \sum_{j=1}^{J} \frac{(d_j - Np_j)^2}{Np_j}$$

- Log-likelihood ratio:

$$G^2 = 2\sum_{j=1}^{J} d_j \log\left(\frac{d_j}{Np_j}\right)$$

where J is the number of categories, N is the total number of responses, $d_1, ..., d_J$ are the number of responses for each category and $p_1, ..., p_J$ are predicted response *probabilities* for each category.

Computing the χ^2 and G^2 cost example

For example, 2 classes (j=1 or j=2), N=5 observations:

$$d_1 = 2, d_2 = 3$$

Assume our model gives predictions:

$$p_1 = .3, p_2 = .7$$

We get

$$\chi^2 = .1905$$
$$G^2 = .23$$

Computing the χ^2 and G^2 cost example

For example, 2 classes (j=1 or j=2), N=5 observations:

$$d_1 = 2, d_2 = 3$$

Assume our model gives predictions:

$$p_1 = .3, p_2 = .7$$

We get

$$\chi^2 = .1905$$
$$G^2 = .23$$

- Find the values of p_1 and p_2 that minimize χ^2 and G^2

Computing the χ^2 and G^2 cost example

For example, 2 classes (j=1 or j=2), N=5 observations:

$$d_1 = 2, d_2 = 3$$

Assume our model gives predictions:

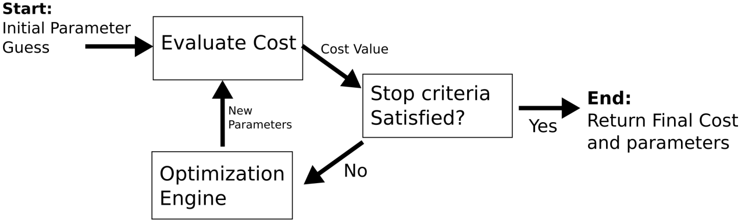
$$p_1 = .3, p_2 = .7$$

We get

$$\chi^2 = .1905$$
$$G^2 = .23$$

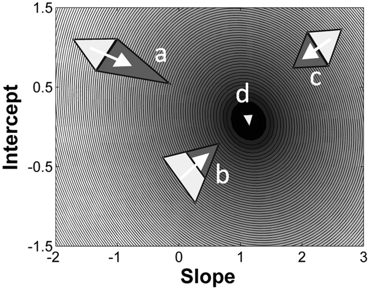
- Find the values of p_1 and p_2 that minimize χ^2 and G^2
- Different cost functions can lead to different costs even when they are minimized. The parameters that minimize these costs are the same because they have the same global minimum.

- So far, we've seen examples of cost functions and examples of models. Parameter estimation also requires an optimizer.
- The optimizer is a method that searches the parameter space to minimize the cost. Several techniques are possible. Generally, but not always it involves an iterative algorithm:



- Which algorithm to use depends on the nature and assumptions made in the model.

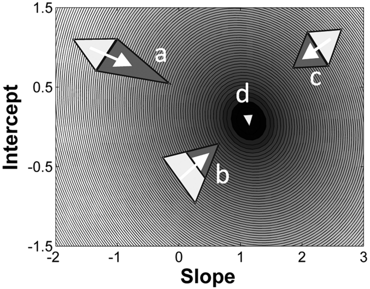
Example: Nelder-Mead "Simplex Method" for 2 parameters



1. A simplex is a geometrical figure that consists of an arbitrary number of interconnected points in an arbitrary number of dimensions.

https://www.youtube.com/watch?v=j2gcuRVbwR0

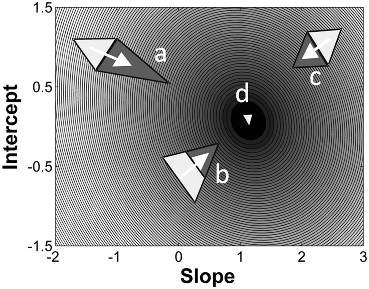
Example: Nelder-Mead "Simplex Method" for 2 parameters



1. A simplex is a geometrical figure that consists of an arbitrary number of interconnected points in an arbitrary number of dimensions.
2. The simplex method evaluates the cost at every point of the simplex

https://www.youtube.com/watch?v=j2gcuRVbwR0

Example: Nelder-Mead “Simplex Method” for 2 parameters



1. A simplex is a geometrical figure that consists of an arbitrary number of interconnected points in an arbitrary number of dimensions.
2. The simplex method evaluates the cost at every point of the simplex
3. The point with the worst fit (largest cost) is displaced (white arrows a and b) or contracted (white arrow b)

https://www.youtube.com/watch?v=j2gcuRVbwR0

Important Assumptions in Parameter Estimation and Algorithms to Solve them

The simplex algorithm is very general, but it can be very slow and get stuck.

Other algorithms can improve parameter estimation, but make certain assumptions on the model. We focus on the following two:

Important Assumptions in Parameter Estimation and Algorithms to Solve them

The simplex algorithm is very general, but it can be very slow and get stuck.

Other algorithms can improve parameter estimation, but make certain assumptions on the model. We focus on the following two:

- **Linearity:** whether the fitted function is linear in the parameters, *e.g.* $y = \alpha + \beta x$ is linear, but $y = a(bt + 1)^{-c}$ is not. Linearity can be sometimes recovered with algebraic manipulations. Linear problems are very easy to solve (e.g. linear regression).

Important Assumptions in Parameter Estimation and Algorithms to Solve them

The simplex algorithm is very general, but it can be very slow and get stuck.
Other algorithms can improve parameter estimation, but make certain assumptions on the model. We focus on the following two:

- **Linearity:** whether the fitted function is linear in the parameters, *e.g.* $y = \alpha + \beta x$ is linear, but $y = a(bt + 1)^{-c}$ is not. Linearity can be sometimes recovered with algebraic manipulations. Linear problems are very easy to solve (e.g. linear regression).
- **Differentiability:** whether the fitted function is “smooth”. When a function is smooth, one can efficiently calculate the direction and magnitude of the new parameters (*e.g.* Neural networks).

Important Assumptions in Parameter Estimation and Algorithms to Solve them

The simplex algorithm is very general, but it can be very slow and get stuck.
Other algorithms can improve parameter estimation, but make certain assumptions on the model. We focus on the following two:

- **Linearity:** whether the fitted function is linear in the parameters, *e.g.* $y = \alpha + \beta x$ is linear, but $y = a(bt+1)^{-c}$ is not. Linearity can be sometimes recovered with algebraic manipulations. Linear problems are very easy to solve (e.g. linear regression).
- **Differentiability:** whether the fitted function is "smooth". When a function is smooth, one can efficiently calculate the direction and magnitude of the new parameters (*e.g.* Neural networks).

If none of these assumptions are true, then general search algorithms such as simplex must be used.

Problems with Parameter Estimation

- If the number of parameters is very large, optimization can be very slow

Problems with Parameter Estimation

- If the number of parameters is very large, optimization can be very slow
- The error surface has multiple minima

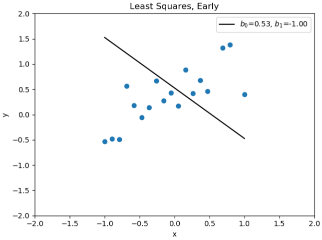
Problems with Parameter Estimation

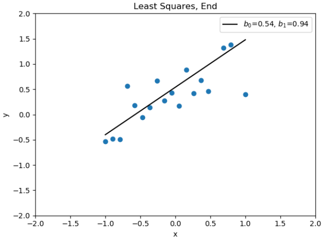
- If the number of parameters is very large, optimization can be very slow
- The error surface has multiple minima
- The error surface can be very “bumpy”

Problems with Parameter Estimation

- If the number of parameters is very large, optimization can be very slow
- The error surface has multiple minima
- The error surface can be very “bumpy”
- Except for certain Bayesian or Bootstrapping techniques, there is no measure of confidence over the parameters

Example: Fitting a linear model





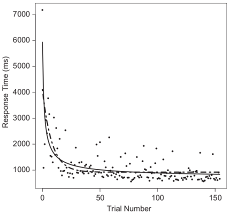
$$y_n = \alpha + \beta x_n + \epsilon_n$$

- True parameters: $\alpha = .5, \beta = 1.0$, i.e. $y_n = .5 + 1.0x_n + e_n$
- Fitted parameters: $\alpha = 0.47, \beta = 1.18$
- The final value of the RMSE is .113. This means, on average, the data points are .113 apart from the prediction.
- Using the `scipy.optimize.minimize` function

For the following examples seen, what cost function can you use (if any)? Is the cost function linear and/or “smooth” in its parameters?

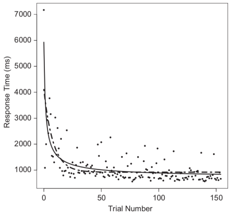
- Skill acquisition power law
- Skill acquisition exponential law
- Memory recall
- BOLD signal with fMRI
- Sequential Sampling Model (with and without trial-to-trial variability)

Skill acquisition



- Exponential (dashed): $RT = e^{-\alpha N}$
- Power (solid): $RT = N^{-\beta}$

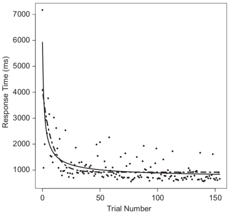
Skill acquisition



- Exponential (dashed): $RT = e^{-\alpha N}$
- Power (solid): $RT = N^{-\beta}$

- Exponential and Power function above have similar properties

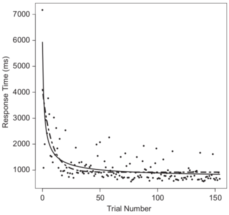
Skill acquisition



- Exponential (dashed): $RT = e^{-\alpha N}$
- Power (solid): $RT = N^{-\beta}$

- Exponential and Power function above have similar properties
- Measured RT is a real number $\rightarrow$ RMSE

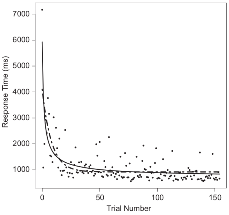
Skill acquisition



- Exponential (dashed): $RT = e^{-\alpha N}$
- Power (solid): $RT = N^{-\beta}$

- Exponential and Power function above have similar properties
- Measured RT is a real number $\rightarrow$ RMSE
- RT is not a linear function of α or β (but it can be manipulated so that it becomes linear).

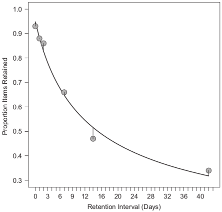
Skill acquisition



- Exponential (dashed): $RT = e^{-\alpha N}$
- Power (solid): $RT = N^{-\beta}$

- Exponential and Power function above have similar properties
- Measured RT is a real number $\rightarrow$ RMSE
- RT is not a linear function of α or β (but it can be manipulated so that it becomes linear).
- RT is differentiable (it is a smooth function of its parameters)

Example Model: Memory recall example

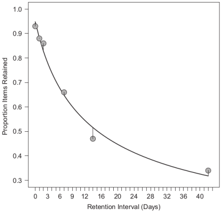


For example: Predicted probability of recall, as a function of time:

$$p(t) = a(bt + 1)^{-c}$$

- $p(t) = \frac{d}{N}$ is the proportion of correct responses.

Example Model: Memory recall example

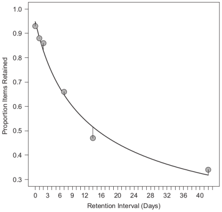


For example: Predicted probability of recall, as a function of time:

$$p(t) = a(bt+1)^{-c}$$

- $p(t) = \frac{d}{N}$ is the proportion of correct responses.
- Measured responses is a two category reponse (True or False) $\rightarrow \chi^2$ or G^2

Example Model: Memory recall example

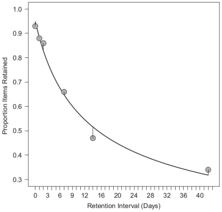


For example: Predicted probability of recall, as a function of time:

$$p(t) = a(bt + 1)^{-c}$$

- $p(t) = \frac{d}{N}$ is the proportion of correct responses.
- Measured responses is a two category reponse (True or False) $\rightarrow \chi^2$ or G^2
- $p(t)$ is not a linear function of α or β (and it can not be manipulated so that it becomes linear).

Example Model: Memory recall example

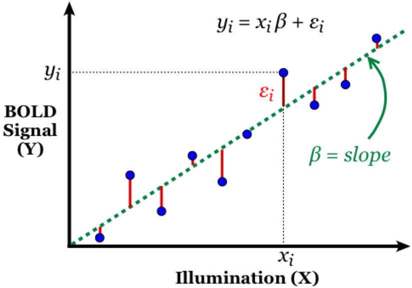


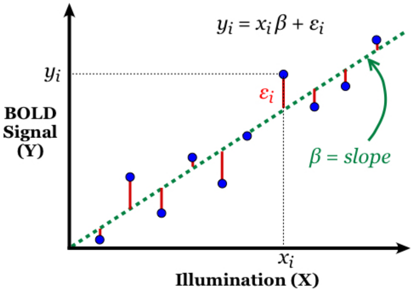
For example: Predicted probability of recall, as a function of time:

$$p(t) = a(bt + 1)^{-c}$$

- $p(t) = \frac{d}{N}$ is the proportion of correct responses.
- Measured responses is a two category reponse (True or False) $\rightarrow \chi^2$ or G^2
- $p(t)$ is not a linear function of α or β (and it can not be manipulated so that it becomes linear).
- $p(t)$ is differentiable (it is a smooth function of its parameters)

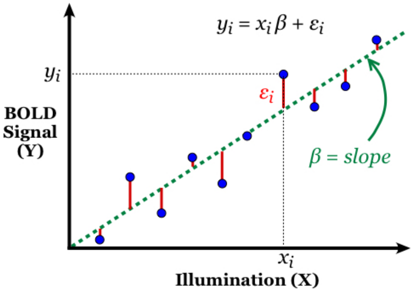
Example Model: BOLD signal in fMRI





- This is the same case as the previous example
  $y_n = \alpha + \beta x_n + e_n$

Example Model: BOLD signal in fMRI



- This is the same case as the previous example $y_n = \alpha + \beta x_n + e_n$
- It is linear and differentiable

Fit the sequential sampling model to match the reaction times (RT).

Without trial-to-trial variability

With trial-to-trial variability

Fit the sequential sampling model to match the reaction times (RT).

Without trial-to-trial variability

- There exists an approximate closed form solution for RT (see slide 23 lecture01.pdf).

With trial-to-trial variability

Fit the sequential sampling model to match the reaction times (RT).

Without trial-to-trial variability

- There exists an approximate closed form solution for RT (see slide 23 lecture01.pdf).
- RT are continuous, so RMSE

With trial-to-trial variability

Fit the sequential sampling model to match the reaction times (RT).

Without trial-to-trial variability

- There exists an approximate closed form solution for RT (see slide 23 lecture01.pdf).
- RT are continuous, so RMSE
- It is non-linear

With trial-to-trial variability

Fit the sequential sampling model to match the reaction times (RT).

Without trial-to-trial variability

- There exists an approximate closed form solution for RT (see slide 23 lecture01.pdf).
- RT are continuous, so RMSE
- It is non-linear
- It is differentiable

With trial-to-trial variability

Fit the sequential sampling model to match the reaction times (RT).

Without trial-to-trial variability

- There exists an approximate closed form solution for RT (see slide 23 lecture01.pdf).
- RT are continuous, so RMSE
- It is non-linear
- It is differentiable

With trial-to-trial variability

- There exists no closed form function describing RT.

Fit the sequential sampling model to match the reaction times (RT).

Without trial-to-trial variability

- There exists an approximate closed form solution for RT (see slide 23 lecture01.pdf).
- RT are continuous, so RMSE
- It is non-linear
- It is differentiable

With trial-to-trial variability

- There exists no closed form function describing RT.
- The cost function cannot be formulated but it can be evaluated.

Fit the sequential sampling model to match the reaction times (RT).

Without trial-to-trial variability

- There exists an approximate closed form solution for RT (see slide 23 lecture01.pdf).
- RT are continuous, so RMSE
- It is non-linear
- It is differentiable

With trial-to-trial variability

- There exists no closed form function describing RT.
- The cost function cannot be formulated but it can be evaluated.
- Only Simplex or other search algorithm can be used.