Models of Categorisation and Memory

Mix of model description and R implementation

First:

- `optim` to minimise summed squared error between a model's predictions and observed data
- Psychophysics; exemplar model of binary categorisation

Around lunch:

- The SIMPLE model of memory; `optimize`
- Maximum likelihood (Simon)

g.d.a.brown@warwick.ac.uk

Models of Memory & Categorisation

Aims of session:

Implement models; play around with them to get a feel for their behaviour – develop intuitions, get some practice

- Simple psychophysical models
- Use `optim` to minimize model error
- Describe and implement basic models of categorisation
- Describe and implement basic model of memory

R Introduction

Nathaniel D. Phillips

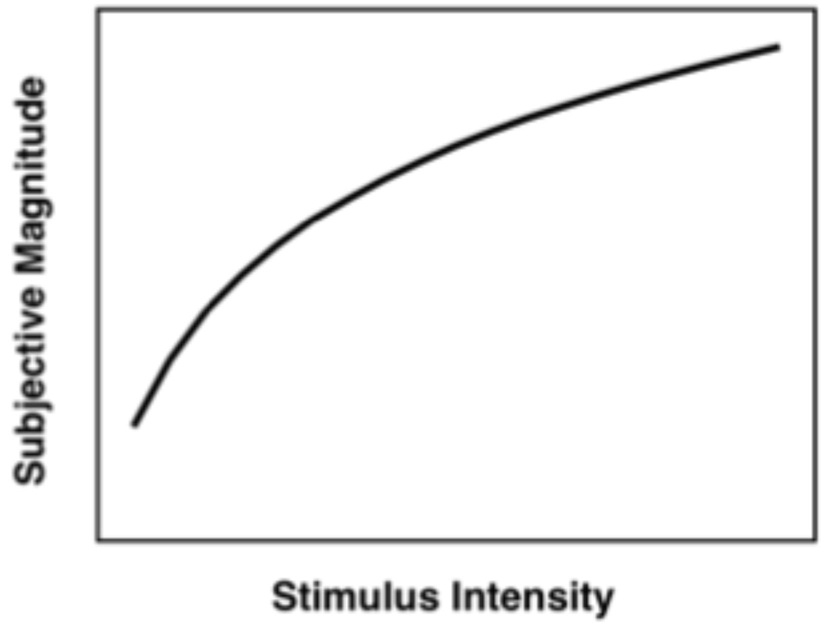
YaRrr! The Pirate's Guide to R

(free e-book)

`nathanieldphillips.com/thepiratesguidetor/`

A Basic Psychophysical Function

How does the subjective magnitude of a stimulus increase as a function of actual stimulus magnitude?

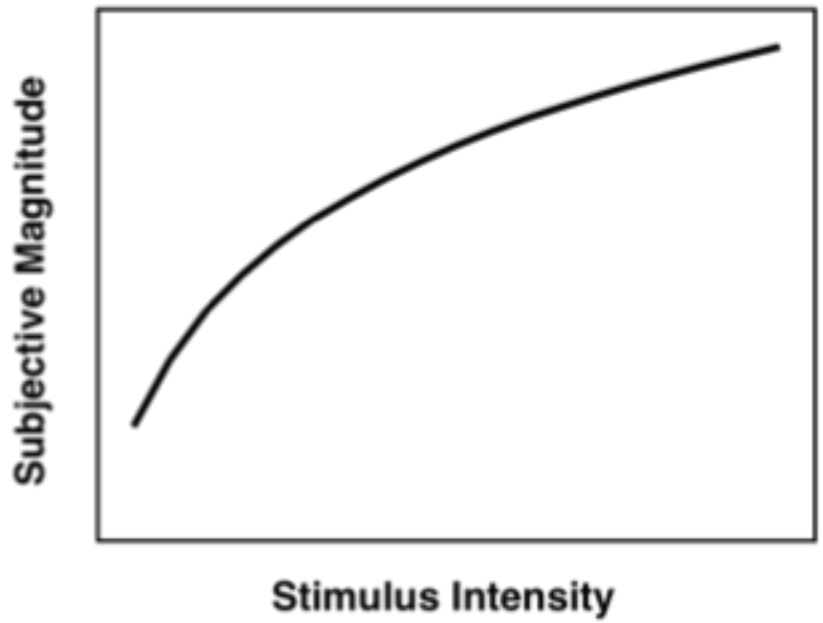


Stevens' Law: Subjective magnitude is a power function of actual stimulus magnitude:

- $S = a * I^b$
- Where: S = subjective magnitude; I is stimulus intensity; *a* and *b* are constants (free parameters)

A Basic Psychophysical Function

How does the subjective magnitude of a stimulus increase as a function of actual stimulus magnitude?



Stevens' Law: Subjective magnitude is a power function of actual stimulus magnitude:

- S = a* I^{b}
- Where: S = subjective magnitude; I is stimulus intensity; *a* and *b* are constants (free parameters)

First Task:

stimuli = [2 4 6 8 10 12]

Write a function that produces Stevens' Law predictions for any set of stimuli and any values of the parameters *a* and *b.*

Apply the function to the vector of stimuli above, with (e.g.) *a* = 1 and *b* = .5

[Remember the form of a function:]

```
NAME <- function(INPUTS) {
  ACTIONS
  return(OUTPUT)
}
```

First Task: Solution 1

stimuli = [2 4 6 8 10 12]

Write a function that produces Stevens' Law predictions for any set of stimuli (e.g., the vector above) and any values of the parameters *a* and *b*

```
stevens_fun1 <- function(x, a, b) {
  output = a*x^b
}
#Some stimuli (stimulus magnitudes) for the input
stimuli<-c(2, 4, 6, 8, 10, 12)

#Find predictions for a=1 and b=.07
predictions <- stevens_fun1(stimuli, 1, 0.7)
```

First Task: Solution 2

stimuli = [2 4 6 8 10 12]

Write a function that produces Stevens' Law predictions for any set of stimuli (e.g., the vector above) and any values of the parameters *a* and *b*

```
#A simpler way to write the function
stevens_fun2 <- function(x, a, b) a*x^b

#Check it produces the same output
predictions2 <- stevens_fun2(stimuli, 1, 0.7)
```

Second Task:

stimuli = [2 4 6 8 10 12]

observations = [1.1 1.5 2.1 2.5 2.7 3.2]

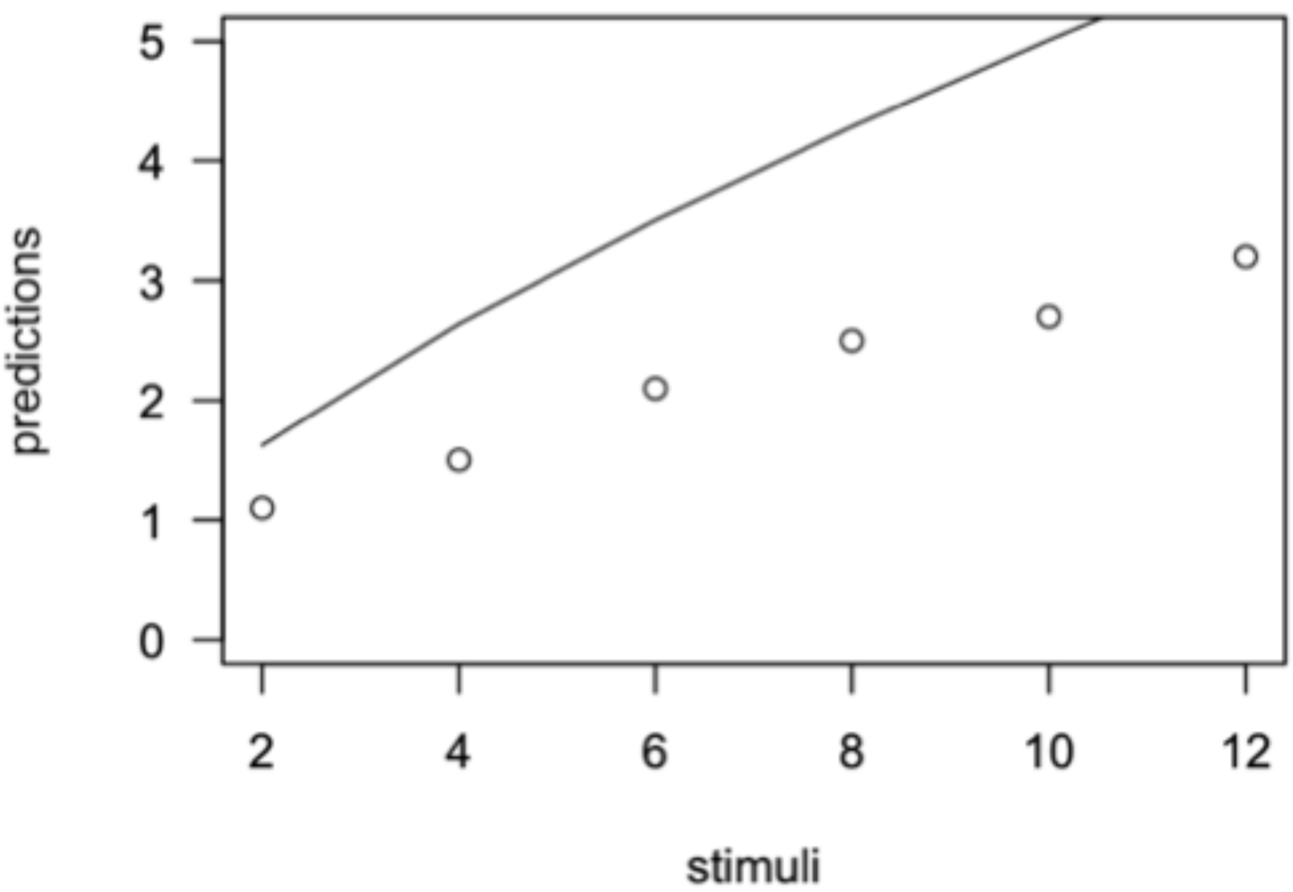
Draw a graph:

- Plot a line graph illustrating the predictions of Stevens' Law, using parameter values *a* = 1 and *b* = 0.7
- Plot (on the same graph) the observed data points above

Second Task: Solution

stimuli = [2 4 6 8 10 12]

observations = [1.1 1.5 2.1 2.5 2.7 3.2]



Second Task: Solution

stimuli = [2 4 6 8 10 12]

observations = [1.1 1.5 2.1 2.5 2.7 3.2]

```
####Draw a line graph of the data (observations) and predictions

plot(stimuli, predictions, type="l", las=1)

####Add points representing some experimental data (observations)

observations <-c(1.1, 1.5, 2.1, 2.5, 2.7, 3.2)

points(stimuli, observations, type="p", las=1, ylab="")
```

Comparing Model and Data

We now have:

stimuli=[2 4 6 8 10 12]

observations= [1.1 1.5 2.1 2.5 2.7 3.2]

What values of *a* and *b* will minimise the deviation between observations (i.e., data) and the predictions of the Stevens' Law model?

We will use **Root Mean Squared Deviation** as our discrepancy measure:

1) Compute the differences (observations-predictions) [note: this is a *vector* of differences; "predictions" is the output from your Stevens function]

2) Square the differences, then take the mean of the squared differences, then take the square root (`sqrt`) of that mean

Third Task:

Write a **function** that does the following:

Takes as input the stimuli, observations (previous exercise) and parameters *a* and *b*

Outputs the RMSD (root mean square deviation) between model (Stevens' Law) and data (observations) for any parameter values

Try various values of the parameters to find the values that minimise RMSD

Third Task: Solution

Notice we pass the parameters as a two-element vector V

```
rmsd = function (V,stimuli,observations) {
  a=V[1]
  b=V[2]
  predictions <- stevens_fun1(stimuli, a, b) #Using the function we made
  rmsd_error=sqrt(mean((observations-predictions)^2)) #Done on whole vectors
  return(rmsd_error)
}
```

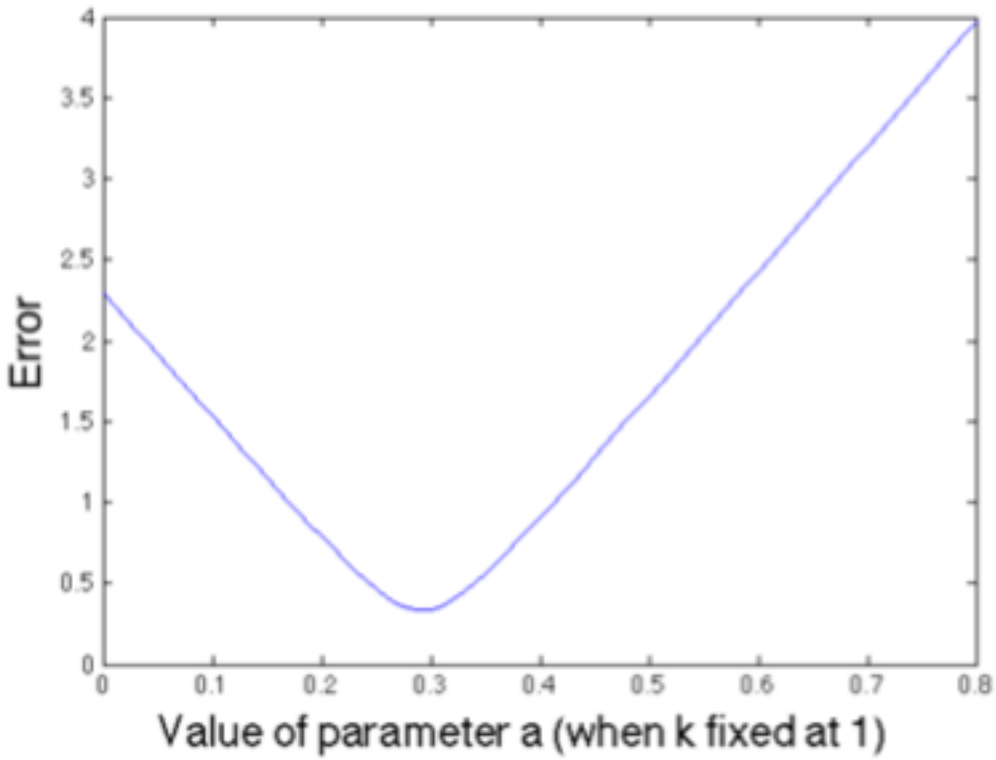
So to call the function:

```
#Check to see if it works
rmsd(c(1,.7),stimuli,observations)
```

The error surface

How can we calculate the best-fitting values of *a* and *b* without trying out every possible value by hand?

Here is an **error surface** when *b* = 1 and *a* varies:



The `optim` command

The command `optim` will do this for us.

We already have a **function** that will take *a* and *b* as input and output the corresponding summed squared error

We can now apply `optim` to this function

Function Minimisation

The `optim` function does fairly simple gradient-descent minimisation (various options)

It may find “local minima” so try different starting points

```
results<-optim(c(2,1),rmsd,stimuli=stimuli,observations=observations )

param_ests<-results$par
error<-results$value
param_ests
error
```

Function Minimisation

The `optim` function does fairly simple gradient-descent minimisation (various options)

It may find "local minima" so try different starting points

Starting values

```
results<-optim(c(2,1),rmsd,stimuli=stimuli,observations=observations )

param_ests<-results$par
error<-results$value
param_ests
error
```

Function Minimisation

The `optim` function does fairly simple gradient-descent minimisation (various options)

It may find "local minima" so try different starting points

```
results<-optim(c(2,1),rmsd,stimuli=stimuli,observations=observations )

param_ests<-results$par
error<-results$value
param_ests
error
```

Name of function whose output must be minimised

Function Minimisation

The `optim` function does fairly simple gradient-descent minimisation (various options)

It may find “local minima” so try different starting points

```
results<-optim(c(2,1),rmsd,stimuli=stimuli,observations=observations)

param_ests<-results$par
error<-results$value
param_ests
error
```

Inputs that “rmsd” needs

Function Minimisation

The `optim` function does fairly simple gradient-descent minimisation (various options)

It may find "local minima" so try different starting points

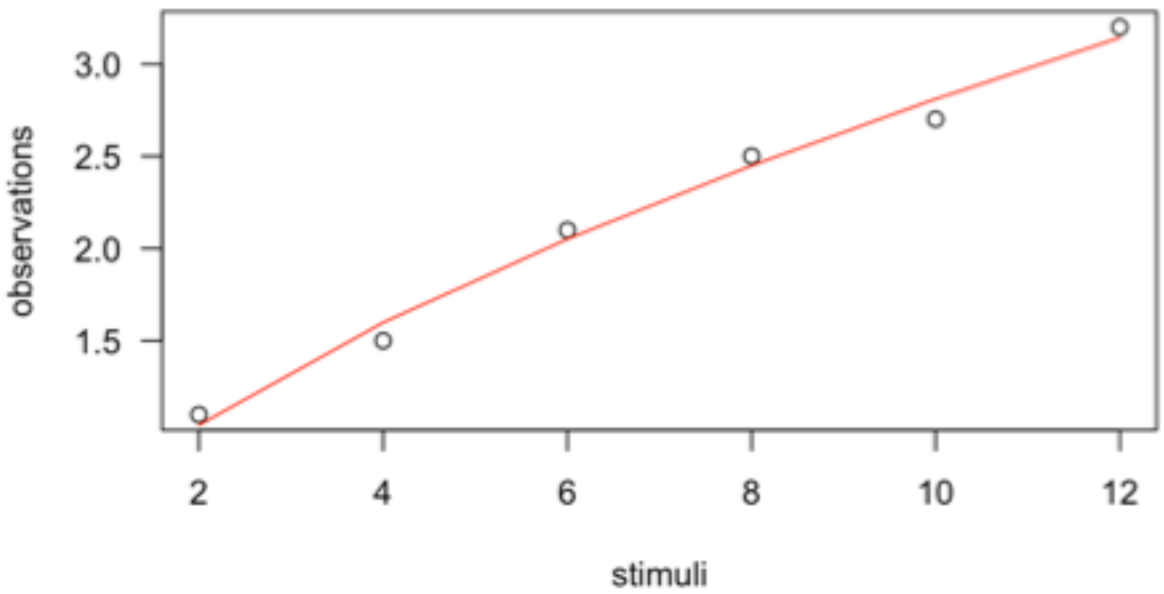
```
results<-optim(c(2,1),rmsd,stimuli=stimuli,observations=observations )

param_ests<-results$par
error<-results$value
param_ests
error
```

parameter estimates are in results

Fourth Task: Use `optim`

Use `optim` to find the values of *a* and *b* that minimise the RMSD error between model and data (same stimuli and responses as before)

Then plot the results on a graph:



Fourth Task: Solution

Use `optim` to find the values of *a* and *b* that minimise the RMSD error between model and data (same stimuli and responses as before)

```
dev.off() #Clear old plot
#First plot the data (observations)
plot(stimuli, observations, type="p",las=1)

#Now we need to calculate predictions with optim's estimated parameters
best_predictions <- stevens_fun1(stimuli, results$par[1], results$par[2])

#Now we can draw the graph
points(stimuli, best_predictions, type="l", col="red", las=1)
```

SHORT BREAK

Categorisation

The ability to place objects into categories seems fundamental to cognition

Categorisation based on properties of objects

- Given the colour and shape of a fruit, is it an apple or a banana?
- Given the age and income of a US citizen, how likely is the person to be Republican or Democrat?

Various possibilities:

- Rules (if round and red, -> apple)
- Prototypes (how similar to typical apple?)
- Exemplar models (to be described)

Categorisation: Reading

The book: For exemplar model of categorisation, see:

- Chapter 4

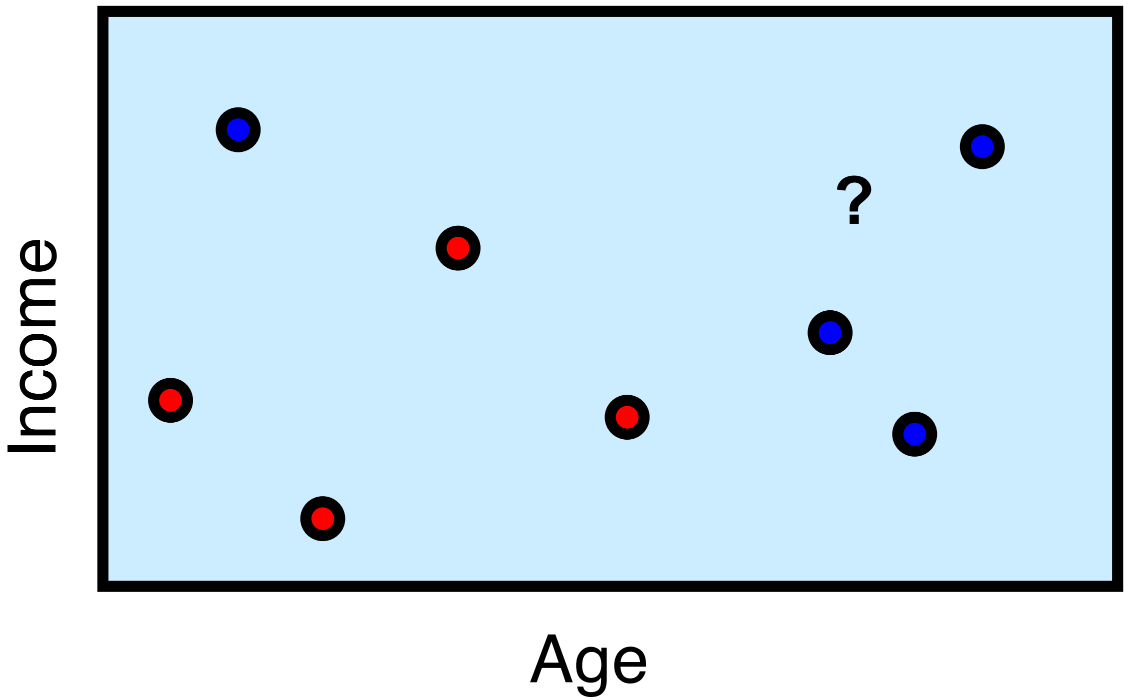
A useful overview:

- Nosofsky, R. M. (1992). Similarity scaling and cognitive process models. *Annual Review of Psychology, 43,* 25-53

Here: just simple binary categorisation of unidimensional stimuli

Categorising a New Item

Suppose you know the age and income of a “new” person but not their political affiliation. Will you place them in the Democrat or Republican category?



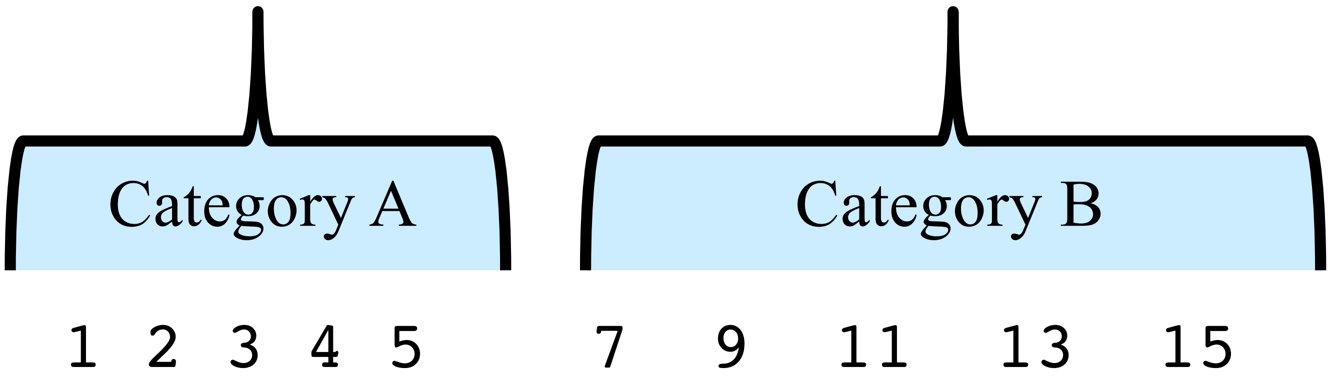
Intuitively:

The new item is similar to Republicans (close in psychological space)

The new item is less similar to Democrats

Binary Categorisation: Single Dimension

Task:

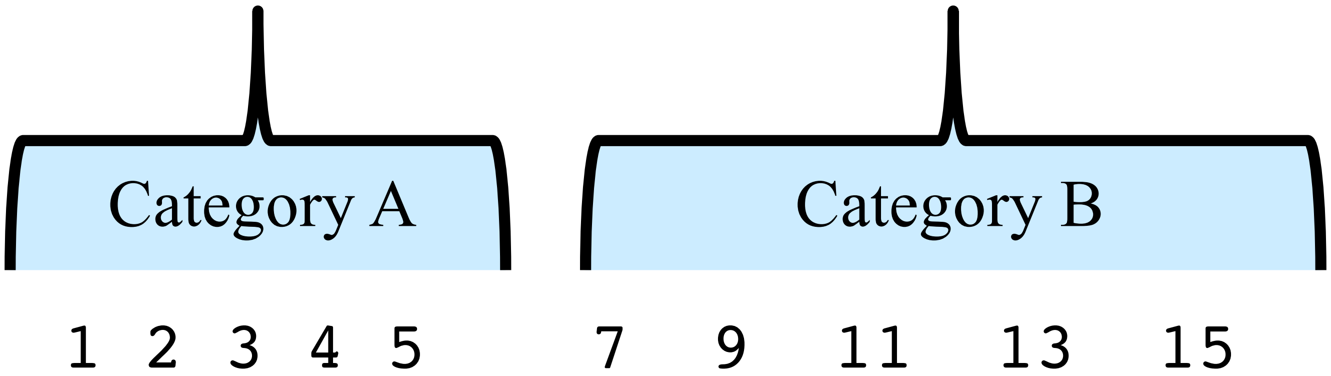


- 10 tones varying in frequency (5 category A; 5 category B)
- Subjects hear tones in random order and must categorise each as “A” or “B”
- Feedback after each category response

Binary Categorisation

Standard exemplar account:

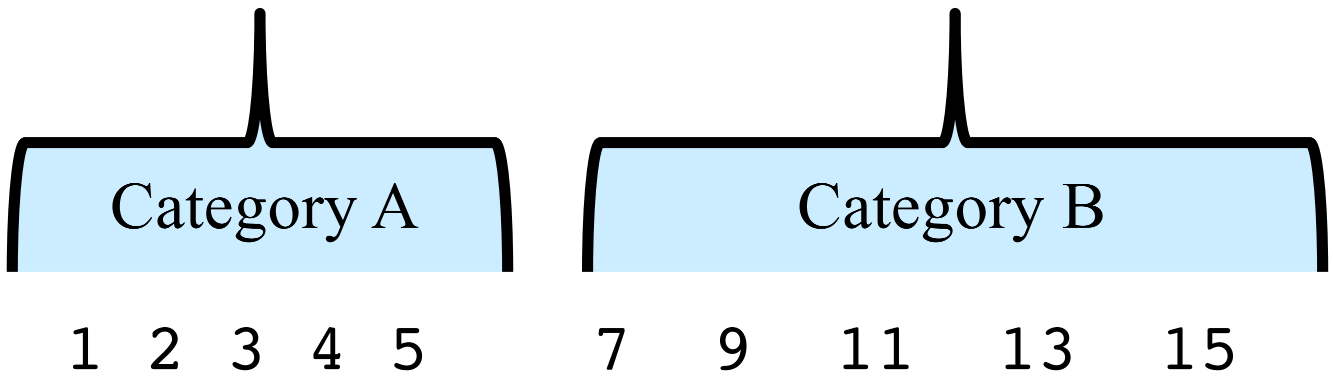
- When each tone is heard, compute its summed similarity to all Category A tones and also its summed similarity to all Category B tones; respond accordingly



Calculating Distance

Just as it sounds:

- Imagine a new tone with a value of 4.5
- Distances to Category A exemplars are:
- 3.5, 2.5, 1.5, .5, .5
- Distances to Category B exemplars are:
- 2.5, 4.5, 6.5, 8.5, 10.5



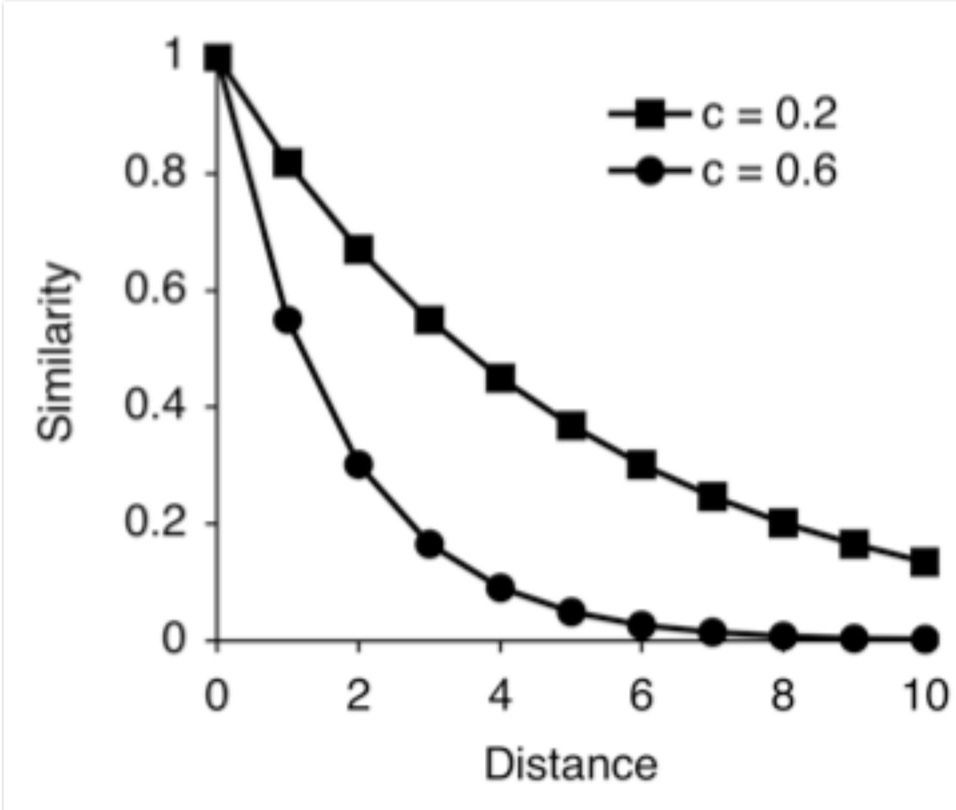
Distance and Similarity

We have an easy way of measuring the distance $d_{i,j}$ between the representations of two locations in psychological space

How do we turn this into a measure of the *similarity* between two locations?

Similarity must *reduce* as distance *increases*:

$$\eta_{i,j} = e^{-c.d_{i,j}}$$



Notes:

1) *c* is a free parameter that governs that rate at which similarity reduces with distance

2) `exp(-c*d)` is R function

The Choice Rule

What is the probability of responding "Category A" to item *i* ?

(summed similarity to Category A items)

divided by

(summed similarity to Category A items plus summed similarity to Category B items)

$$P(A|i) = \frac{\sum_{j \in A} \eta_{i,j}}{\sum_{j \in A} \eta_{i,j} + \sum_{j \in B} \eta_{i,j}}$$

Task 1: To Implement

Write a function to calculate, for any value of parameter *c*, the probability of categorising any stimulus value as category A.

If time, draw a graph to illustrate for stimulus values 1:30

Assume the existing category members are as on previous slide, i.e.,

```
CatA_vals=c(1,2,3,4,5)
CatB_vals=c(7,9,11,13,15)
```

Hints:

- for thestimulus value x, calculate vectors giving the distance between x and Category A members and the distance between x and Category B members
- transform these distances into similarities
- calculate the summed similarities and apply the choice rule

```
#Provide the stimulus values of Category A items
CatA_vals=c(1,2,3,4,5)
#Provide the stimulus values of Category B items
CatB_vals=c(7,9,11,13,15)

#Now we need a function that takes any x value, and param c, and calcula
#categorising x as category A

#We will write this as a function right away
categ = function (c, x) {
dists_to_A=abs(x-CatA_vals) #first calc the distances between x and all
dists_to_B=abs(x-CatB_vals) #then distances between x and category B ite
sims_to_A=exp(-c*dists_to_A) #turn distances into similarities
sims_to_B=exp(-c*dists_to_B) #turn distances into similarities
prob_A=sum(sims_to_A)/(sum(sims_to_A)+sum(sims_to_B))
return(prob_A)
}

#Check that it works
categ(0.5, 2)
```

Task 2: To Implement

Draw a graph to illustrate, for stimulus values 1:30, the probability that each stimulus will be categorised as A

Explore effects of varying the *c* parameter

Hints:

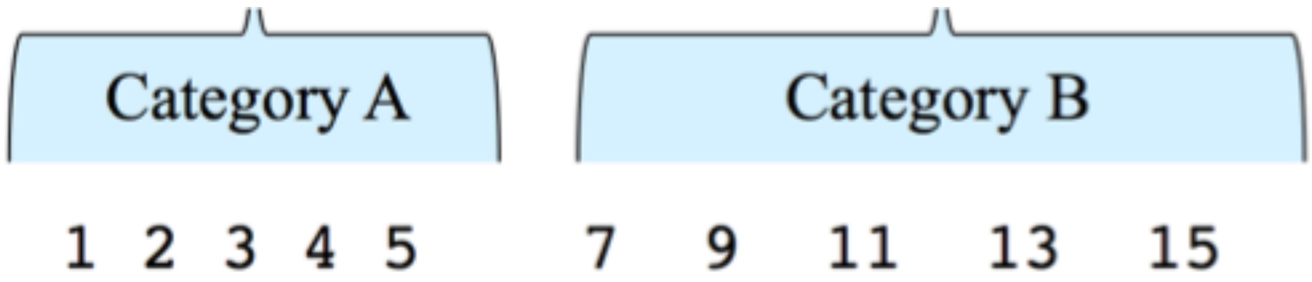
- you can use a for loop for this

Task 2: Solution

```
#Now let us take several stimuli and (for each) return the probabi
#that it will be classified as Category A
values=seq(1,30,.1)

prob_A=NULL #initalise a matrix that we will save results into
c=0.5 #the similarity-distance parameter
for (n in 1:length(values)){
prob_A[n]=categ(c,values[n])
}

#And plot the values
dev.off()
plot(values, prob_A, las=1)
```

Some Issues

1) Take a stimulus equidistant between the two categories (6)

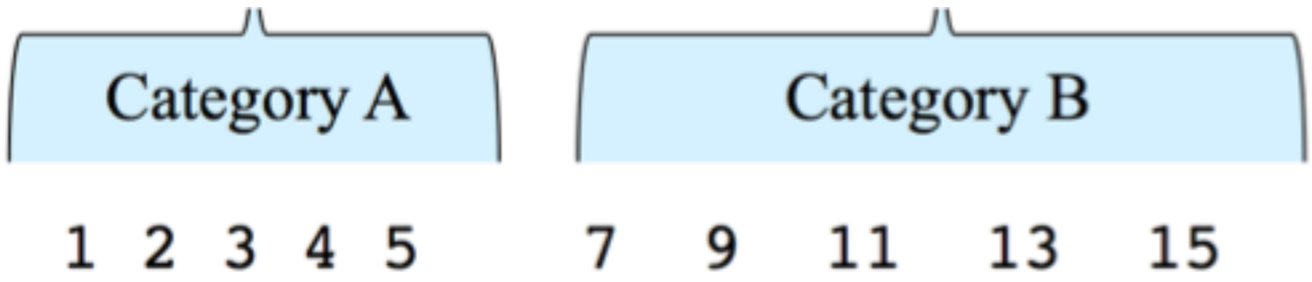


Is it categorised as A or B?

What can we conclude from the result?

Some Issues

1) Take a stimulus equidistant between the two categories (6)



Is it categorised as A or B?

What can we conclude from the result?

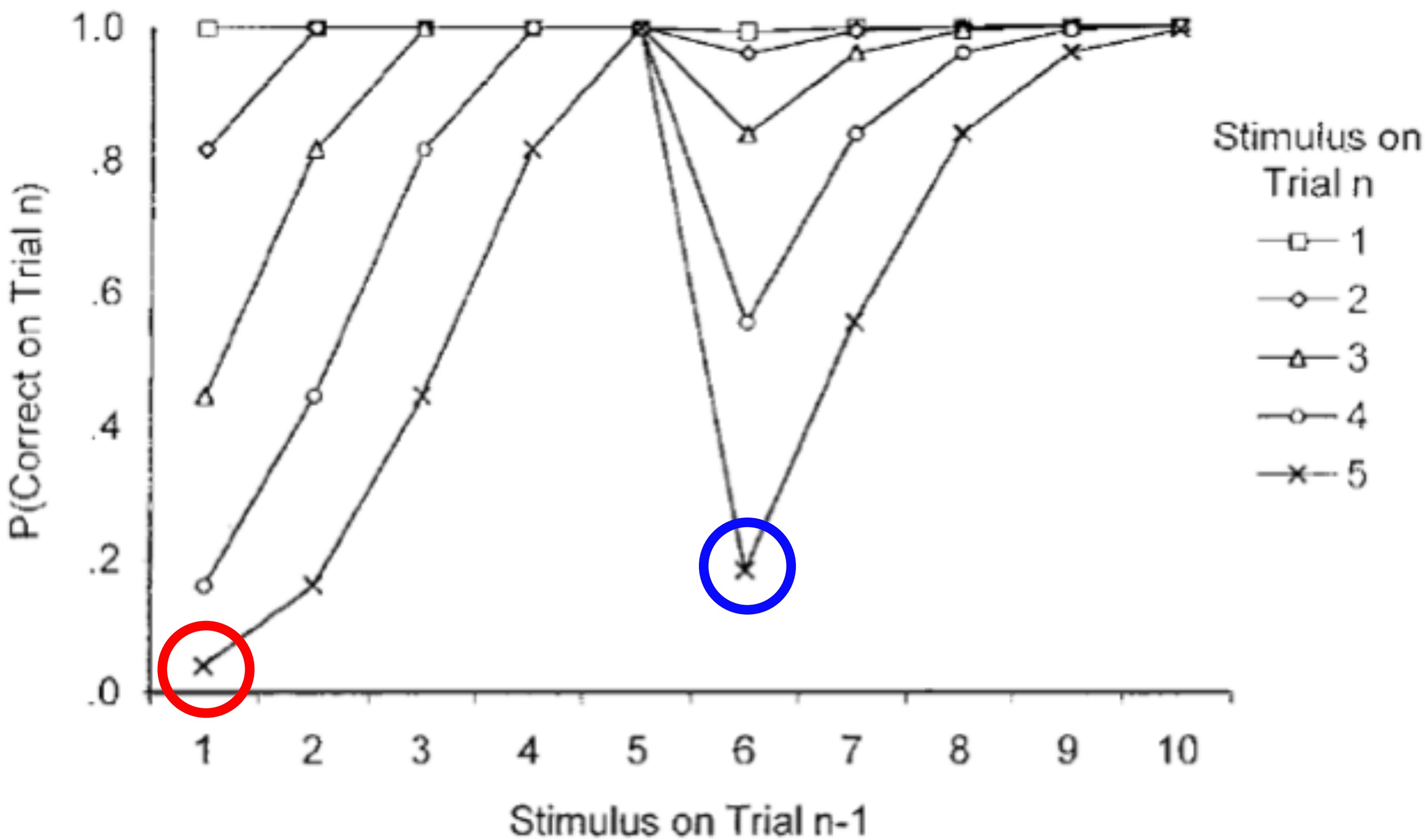
2) Borderline stimuli may be misclassified (e.g. if tone 5 is preceded by tone 1 it may be wrongly categorised as a “B”)

Suggests very different models

END

Other Evidence

Borderline stimuli may be mis-classified (e.g. if tone 5 is preceded by tone 1 it may be wrongly categorised as a “B”)



Alternative Approaches

Models like the GCM assume that:

- Items in memory can be seen as having point locations in some fixed absolute space*
- Similarities are used in the computations

But:

- Much work in psychophysics suggests people have no long-term stable representations of the absolute magnitudes of stimuli (e.g. loudnesses)

Alternative model:

- Assume people have access only to *differences between successive stimuli*

*Although note only *distances between* items used in model

More generally….

Now a variety of relative models (that assume no fixed stable internal magnitude representations)…e.g.…

- The MAC (Memory and Contrast) model of ***binary categorisation*** (e.g. Stewart, Brown, & Chater, 2002)
- The RJM (Relative Judgment Model) of ***absolute identification*** (Stewart, Brown, & Chater, 2005)
- The DbS (Decision by Sampling) model of ***judgement and choice*** (e.g. Stewart, Chater, & Brown, 2006)

“Relative Judgment” Models

- Some recent models of categorisation, absolute identification, and judgement assume that participants can only make *relative judgements*
- People categorise/identify/evaluate stimuli by comparing them with recent stimuli whose identity or category label is known

S_{n-1} S_n

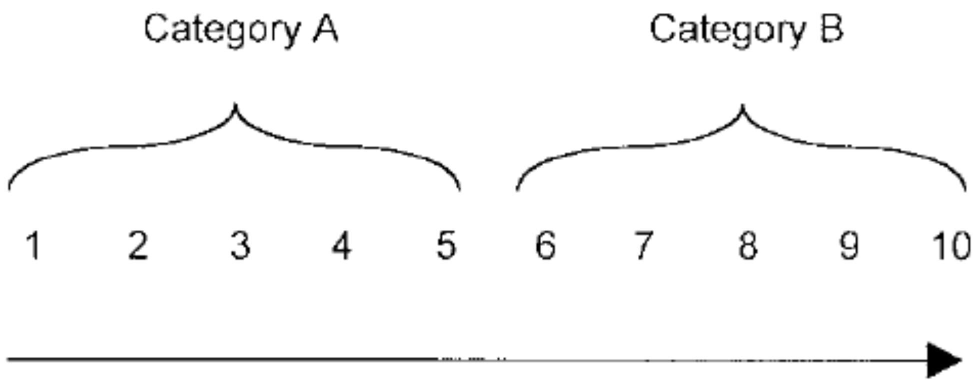
1 2 3 4 5 6 7 <u>8</u> 9 10

Current stimulus to be judged is S_n; previous was 3; difference is + 5

The Memory and Contrast Model

Absent stable and accurate LTM representations of stimulus magnitudes, assume that participants compare current stimulus just with previous stimulus

If the difference is greater than a threshold, change response category



EXAMPLE:

If $S_{n-1} = 3$; threshold=3.5:

If $S_n >= 7$; respond B

The Next Task

Implement a simple version of the MAC model and show how it reproduces the basic finding shown above (see next slide for steps)

If additional time:

- Explore ways to take into account stimuli from further back in the sequence (i.e., S_{n-2} as well as S_{n-1})

- Consider how to extend to absolute identification (each stimulus has its own label)

Steps for the model..

— Assume 20 stimuli: `stimuli = [1:20]`

— First 10 are "A"s, largest 10 are "B"s

— Consider every stimulus (S_n) being preceded by every other stimulus (S_{n-1})

— If difference between S_n and S_{n-1} is greater than a threshold, then switch response category if difference is in appropriate direction

— Show performance on each stimulus for various values of threshold

END