Package ‘JoF’

October 12, 2022

Type Package

Title Modelling and Simulating Judgments of Frequency

Version 0.1.0

Maintainer Markus Burkhardt <markus.burkhardt@psychologie.tu-chemnitz.de>

Description In a typical experiment for the intuitive judgment of frequencies (JoF) different stimuli with different frequencies are presented. The participants consider these stimuli with a constant duration and give a judgment of frequency. These judgments can be simulated by formal models: PASS 1 and PASS 2 based on Sedlmeier (2002, ISBN:978-0198508632), MINERVA 2 baesd on Hintzman (1984) <doi:10.3758/BF03202365> and TODAM 2 based on Murdock, Smith & Bai (2001) <doi:10.1006/jmps.2000.1339>. The package provides an assessment of the frequency by determining the core aspects of these four models (attention, decay, and presented frequency) that can be compared to empirical results.

Depends R (>= 3.1.0)

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.0.2

Suggests testthat, knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Markus Burkhardt [aut, cre]

Repository CRAN

Date/Publication 2020-04-03 14:10:36 UTC

R topics documented:

`MINERVA2` *Modeling Judgments of Frequency with MINERVA 2*

Description

Modeling Judgments of Frequency with MINERVA 2

Usage

```
MINERVA2(x, y, ..., sqc, L, dec = NULL)
```

Arguments

`x`	input handled by MINERVA 2. Values -1, 0 and 1 are allowed. -1 represents the absence of a feature, 0 the irrelevance of a feature and 1 the presence of a feature.
`y`	another input handled by MINERVA 2. At least two inputs are needed for the simulation.
`...`	other inputs for modeling.
`sqc`	sequence of the different objects. Each input gets an ascending number. `x` gets the value `1`, `y` gets the value `2`, `...` gets the value `3` and so on. The argument `sqc = c(1, 2, 3, 2)` means: first input `x` is processed, second input `y` is processed followed by processing input number three and fourth, th input `y` is used again. So `sqc` contains the frequency information too. In `c(1, 2, 3, 2)`, `x` and the third input are presented once. The input `y` is presented twice.
`L`	learning parameter. This is the proportion of a correctly stored vector. `L = 1` means 100 % of the input is processed correctly. If `L` is a vector, each input could be handled differently. So `L = c(.5, .6, .9)` means, input `x` is correctly stored to 50 %, input `y` is stored to 60 % and the third input (inserted in `...`) is stored with 90 % probability.
`dec`	decay is not part of the original version of MINERVA 2. This is just implemented for a better comparison with the other models of JoF. In `dec = NULL`, decay has no effect. For `dec = 'curve'` decay uses a forgetting curve. If dec is a numeric Vector e. g. `dec = c(.8, .9, 1)` the memory traces are weighted. The first represented trace is weighted by .8 the second by .9 and the youngest trace by 1. The value `dec = 1` corresponds with the original model.

Details

Calculations of MINERVA 2 contain four steps.

$$S_i = \frac{\sum_{j=1}^{N} P_j T_{ij}}{N_i}$$

$$A_i = S_i^3$$

$$I = \sum_{i=1}^{M} A_i$$

$$relativeJoF = \frac{I_j}{\sum_j^K I_j}$$

Value

MINERVA2 returns the relative judgment of frequency

References

Dougherty, M. R., Gettys, C. F., & Ogden, E. E. (1999). MINERVA-DM: A memory processes model for judgments of likelihood. *Psychological Review, 106*(1), 180.

Hintzman, D. L. (1984). MINERVA 2: A simulation model of human memory. *Behavior Research Methods, Instruments, and Computers, 16,* 96–101.

Examples

```
#This example is presented in Dougherty,
#Gettys, & Ogden, 1999 (p. 185)
H1 <- c(-1, 1, 0, 1, 0, -1, 1, -1, 0)
H2 <- c(-1, 0, 0, 1, 0, 0, 1, 0, 0)
x <- MINERVA2(H1, H2, sqc = c(2, 1), L = 1)
```

PASS1 *Modeling Judgments of Frequency with PASS 1*

Description

Modeling Judgments of Frequency with PASS 1

Usage

```
PASS1(x, y, ..., sqc, att, dec, ifc, rdm_weights = TRUE, noise = 0)
```

Arguments

`x`	input handled by PASS 1. Only binary input is allowed.
`y`	a second binary input handled by PASS 1. At least two inputs are needed for the simulation.
`...`	other binary inputs for modeling.
`sqc`	sequence of the different objects. Each input gets an ascending number. `x` gets the value `1`, `y` gets the value `2`, `...` gets the value `3` and so on. The argument `sqc = c(1, 2, 3, 2)` means: first input `x` is processed, second input `y` is processed followed by processing input number three and fourth, th input `y` is used again. So `sqc` contains the frequency information too. In `c(1, 2, 3, 2)`, `x` and the third input are presented once. The input `y` is presented twice.

att
: attention is a vector with numeric values between 0 and 1. `att` has the same length like `sqc`, so each input processing have its own value and PASS 1 can modulate attention by time or input. If `att` is exact one numeric value (e.g. `att = .1`), all inputs get the same parameter of attention.

dec
: decay is a vector with numeric values between -1 and 0. `dec` has the same length as `sqc`, so each input processing have its own value and PASS 1 can modulate decay by time. If `dec` is exact one numeric value (e.g. `dec = -.1`), all inputs get the same parameter of decay.

ifc
: interference is a vector with numeric values between -1 and 0. `ifc` must have the same length as `sqc`. So each inputprocessing have its own value and PASS 1 can modulate inference by time. If `ifc` is exact one numeric value (e.g. `ifc = -.1`), all inputs get the same parameter of inference.

rdm_weights
: a logical value indicating whether random weights in the neural network are used or not. If `rdm_weights = FALSE` all network connections are zero at the beginning.

noise
: a proportion between 0 and 1 which determine the number of randome activiated inputunits (hihger numbers indicate higher noise).

Details

PASS 1 is a simple neural pattern associator learning by delta rule.

Learning:

$$ifU_iandU_jareactivated, then\Delta w_{ij} = \Theta_1(1 - w_{ij})$$

Interference:

$$ifeitherU_iorU_jisactivated, then\Delta w_{ij} = \Theta_2 * w_{ij}$$

Decay:

$$ifneitherU_inorU_jisactivated, then\Delta w_{ij} = \Theta_3 * w_{ij}$$

Value

`PASS1` returns the relative judgment of frequency for each input.

References

Sedlmeier, P. (2002). Associative learning and frequency judgements: The PASS model. In P. Sedlmeier, T. Betsch (Eds.), *Etc.: Frequency processing and cognition* (pp. 137-152). New York: Oxford University Press.

Examples

```
o1 <- c(1, 0, 0, 0)
o2 <- c(0, 1, 0, 0)
o3 <- c(0, 0, 1, 0)
o4 <- c(0, 0, 0, 1)
PASS1(o1, o2, o3, o4,
      sqc = rep(1:4, 4:1), att = .1, dec = -.05,
      ifc = -.025, rdm_weights = FALSE, noise = 0)
```

PASS2 *Modelling Judgments of Frequency with PASS 2*

Description

Modelling Judgments of Frequency with PASS 2

Usage

```
PASS2(x, y, ..., sqc, att, n_output_units = "half", rdm_weights = F, noise = 0)
```

Arguments

x	input handled by PASS 2. Only binary input is allowed.
y	a second binary input handled by PASS 1. At least two inputs are needed for the simulation.
...	other binary inputs for modeling.
sqc	sequence of the different objects. Each input gets an ascending number. `x` gets the value `1`, `y` gets the value `2`, `...` gets the value `3` and so on. The argument `sqc = c(1, 2, 3, 2)` means: first input `x` is processed, second input `y` is processed followed by processing input number three and fourth, th input `y` is used again. So `sqc` contains the frequency information too. In `c(1, 2, 3, 2)`, `x` and the third input are presented once. The input `y` is presented twice.
att	attention is a vector with numeric values between 0 and 1. `att` has the same length like `sqc`, so each input processing have its own value and PASS 1 can modulate attention by time or input. If `att` is exact one numeric value (e.g. `att = .1`), all inputs get the same parameter of attention.
n_output_units	number of output units as numeric value. This must be between 1 and the maximum number of input units. `n_output_units = 'half'` determines the half of the input units.
rdm_weights	a logical value indicating whether random weights in the neural network are used or not. If `rdm_weights = FALSE` all network connections are zero at the beginning.
noise	a proportion between 0 and 1 which determines the number of random activated input units (higher numbers indicate higher noise).

Details

PASS 2 uses a competitive learning algorithm, which usually clusters the input as side effect. If weights are equal, the winning unit is chosen randomly, because of this, each simulation is slightly different.

$$ifanoutputuniO_ilosses : \Delta w_{ij} = 0$$

$$ifanoutputuniO_iwins : \Delta w_{ij} = g_w \frac{a_i}{\sum_i^n a_i} - g_w w_{ij}$$

Value

PASS2 returns the relative judgment of frequency for each input.

References

Sedlmeier, P. (2002). Associative learning and frequency judgements: The PASS model. In P. Sedlmeier, T. Betsch (Eds.), *Etc.: Frequency processing and cognition* (pp. 137-152). New York: Oxford University Press.

Examples

```
o1 <- c(1, 0, 0, 0)
o2 <- c(0, 1, 0, 0)
o3 <- c(0, 0, 1, 0)
o4 <- c(0, 0, 0, 1)
PASS2(o1, o2, o3, o4,
      sqc = rep(1:4, 4:1), att = .1, n_output_units = 2,
      rdm_weights = FALSE, noise = 0)
```

---

plot.JoF *plot progress of judgment of frequency simulation*

Description

plot progress of judgment of frequency simulation

Usage

```
## S3 method for class 'JoF'
plot(x, type = "l", ...)
```

Arguments

x	output of JoF simulation
type	"l" for lines, "p" for points, "b" for both
...	further arguments

Value

Displays the judgment of frequency as proportion of all inputs `print.JoF` *Output of judgment of frequecny simulation*

Description

Output of judgment of frequecny simulation

Usage

```
## S3 method for class 'JoF'
print(x, ...)
```

Arguments

`x`	output of JoF simulation
`...`	further arguments

Value

Displays the judgment of frequency as proportion of all inputs

`TODAM2` *Modeling Judgments of Frequency with TODAM 2*

Description

Modeling Judgments of Frequency with TODAM 2

Usage

```
TODAM2(x, y, ..., sqc, gamma = 1, alpha = 1)
```

Arguments

`x`	input handled by TODAM 2. Normal distributed inputs with mean = 0 and sd = 1 / n are allowed. This representation enables discrimination and similarity between different items. See vignette for details.
`y`	another input handled by TODAM 2. At least two inputs are needed for the simulation.
`...`	other inputs handled by TODAM 2.
`sqc`	sequence of the different objects. Each input gets an ascending number. `x` gets the value `1`, `y` gets the value `2`, `...` gets the value `3` and so on. The argument `sqc = c(1, 2, 3, 2)` means: first input `x` is processed, second input `y` is processed followed by processing input number three and fourth, th input `y` is used again. So `sqc` contains the frequency information too. In `c(1, 2, 3, 2)`, `x` and the third input are presented once. The input `y` is presented twice.

`gamma` is the atttention- or learningparameter. Values between 0 and 1 are allowed. 1 represents perfect learning. If `gamma` iis a vector, each input could be handled differently. So `gamma = c(.5, .6, 1)` means, the third input is stored correctly and betther than the `y` better than first input `x`).

`alpha` represents the decay. If `alpha = 1`, the complete memory vector is used (and no forgetting takes place). If `alpha` is an numeric Vector e. g. `alpha = c(.8, .9, 1)`, the memory vector is weighted. The memory for the first input is weaker than the second than the third.

Details

In the original publication TODAM 2 is more complex and has more parameters. Especially the design for the input is a concatenation between item and context. The normal distributed input has a *mean = 0* and *sd = 1/n*. A pragmatic solution to make the models input comparable is to use a binary input like in PASS. There is no explicit argument for noise.

Convolution:

$$F_i^2 = \sum_{i=1} f_i * f_{m-i+1} and m = 2n - 1$$

Memory:

$$M_t = \alpha M_{t-1} + \gamma F_t^2$$

Correlation

$$R_m = \sum_{(i;j) \in S(m)} F_t^2 there S(m)(i;j) | - (n-1)/2 \leq i, j \leq (n-1)/2 and i - j = m$$

References

Murdock, B. B., Smith, D., & Bai, J. (2001). Judgments of frequency and recency in a distributed memory model. *Journal of Mathematical Psychology, 45,* 564–602. https://doi.org/10.1006/jmps.2000.1339

Examples

```
o1 <- c(-0.27, -0.24, -0.24, 0.75)
o2 <- c(-0.06, -0.55,  0.66, -0.06)
o3 <- c(0.04,  0.57, -0.65,  0.04)
o4 <- c(0.73, -0.39, -0.20, -0.14)
TODAM2(o1, o2, o3, o4, gamma = rep(c(0.7, 0.8), 5),
alpha = 0.95, sqc = rep(1:4, 4:1))
```

Index