Package 'priorInferenceIterative'

October 24, 2021

Type Package

Title Prior Inference Iterative

Version 0.1.0

Author Martin V. Butz, Hania Elkersh, Asya Achimova

Maintainer Asya Achimova <asya.achimova@uni-tuebingen.de>

Description This package contains all the functions for executing the iterative prior inference setting. This package was designed to analyze and model data.

License MIT License

Encoding UTF-8

LazyData true

Suggests testthat,
knitr,
rmarkdown

RoxygenNote 7.1.2

Imports

VignetteBuilder knitr

R topics documented:

`determineSpeakerPostListPrefsSimpleRSAWithPriorPref_dep` *Determine speaker's inference of the posterior listener preferences (iterative setting, dependent on trial order)*

Description

Simple RSA (iterative, dependent on trial order)

This function calculates the speaker's posterior guess of the feature value preferences of the listener in the iterative setting. That means how the speaker infers the preferences of the listener based on the object choice.

Usage

```
determineSpeakerPostListPrefsSimpleRSAWithPriorPref_dep(
  currentObjects,
  featureUtt,
  softPrefValue,
  notObeyInst,
  priorPrefAll
)
```

Arguments

`currentObjects`	A vector of three values in `{1,...,27}` specifying the target and the other two objects in the scene. The target is the first object in the vector (`index = 1`).
`featureUtt`	One of the values `{1,2,3}` specifying which feature is uttered (i.e. shape = 1 / texture = 2 / or color = 3).
`softPrefValue`	A parameter value between `[0,infinity)` (The larger the higher the tendency towards uniform liking). Value reflects how categorical the listener's preferences are: **0:** The listener always picks her preferred object. If the listener prefers *red* objects, she will always pick the *red* object in the scene. **infinity:** It is as likely for the listener to pick *green*, *blue* or *red* objects.
`notObeyInst`	Determines the extent to which the instruction of the speaker is obeyed by the listener. (0 = full obedience, infinity = full instruction ignorance). **Example:** **0:** Listener always picks *red* objects following the utterance *"red"*. **infinity:** Listener as likely to pick *green, blue* or *red* objects even if the utterance is *"red"*.
`priorPrefAll`	A vector of length 9. Probability mass over all feature values. Gives a prior preferences distribution over all (nine) feature values.

Value

A vector of length 9. It contains the speaker's inference of the feature value preferences of the listener dependent on the trial order.

`determineSpeakerPostListPrefsSimpleRSAWithPriorPref_indep_pr`
Determine speaker's inference of the posterior listener preferences (iterative setting, independent of trial order) `posterior = (1 - prior rate) x evidence + (prior rate) x prior`

Description

Simple RSA (iterative, independent of trial order (prior rate))

This function calculates the speaker's posterior guess of the feature value preferences of the listener in the iterative setting. That means how the speaker infers the preferences of the listener based on the object choice.

Usage

```
determineSpeakerPostListPrefsSimpleRSAWithPriorPref_indep_pr(
  currentObjects,
  featureUtt,
  softPrefValue,
  notObeyInst,
  priorPrefAll,
  priorRate
)
```

Arguments

`currentObjects`	A vector of three values in `{1,...,27}` specifying the target and the other two objects in the scene. The target is the first object in the vector `(index = 1)`.
`featureUtt`	One of the values `{1,2,3}` specifying which feature is uttered (i.e. shape = 1 / texture = 2 / or color = 3).
`softPrefValue`	A parameter value between `[0,infinity)` (The larger the higher the tendency towards uniform liking). Value reflects how categorical the listener's preferences are: **0:** The listener always picks her preferred object. If the listener prefers *red* objects, she will always pick the *red* object in the scene. **infinity:** It is as likely for the listener to pick *green*, *blue* or *red* objects.
`notObeyInst`	Determines the extent to which the instruction of the speaker is obeyed by the listener. (0 = full obedience, infinity = full instruction ignorance). **Example:** **0:** Listener always picks *red* objects following the utterance *"red"*. **infinity:** Listener as likely to pick *green, blue* or *red* objects even if the utterance is *"red"*.

priorPrefAll	A vector of length 9. Probability mass over all feature values. Gives a prior preferences distribution over all (nine) feature values.
priorRate	This parameter specifies how much the prior information is weighed into the speaker's decision is weighed into the decision of the speaker regarding the feature preferences of the listener.

Value

A vector of length 9. It contains the speaker's inference of the feature value preferences of the listener independent of the trial order.

Examples

```
determineSpeakerPostListPrefsSimpleRSAWithPriorPref_indep_pr(currentObjects, featureUtt,
softPrefValue, notObeyInst, priorPrefAll, priorRate)

output:
[1]  6.666667e-01 6.666667e-101  3.333333e-01  0.000000e+00  0.000000e+00  0.000000e+00
[7]  0.000000e+00  0.000000e+00  0.000000e+00
```

LL1_1_Iterative_dep_notObey0

Cost function for one parameter optimization (iterative setting, dependent on trial order). Optimizing softness. Non-obedience fixed at 0.

Description

Simple RSA

1 parameter optimization; The softness parameter is optimized.

The non-obedience parameter is fixed.

Usage

```
LL1_1_Iterative_dep_notObey0(params, data)
```

Arguments

params	One value vector, which specifies one of two parameters to be optimized: 1. softPrefValue is optimized, i.e. The strength of "preferring one entity over others". (The larger the value the higher the tendency towards uniform liking) 2. non-obedience (default = 0), i.e. The extent to which the instruction of the speaker is obeyed by the listener. (0 = full obedience, infinity = full instruction ignorance)
data	A Matrix with data rows. column structure: [1:OC1,OC2,OC3,4:UUFeat, 5:Q1Feat,6:Q2Feat] [7:Q1AnswerV1,V2,V3, 10:Q2AnswerV1,V2,V3]

1:OC1 Object 1. A value between 1 and 27.

2:OC2 Object 2. A value between 1 and 27.

3:OC3 Object 3. A value between 1 and 27.

4:UUFeat Uttered feature. A number between 1 and 3. (1: shape, 2: pattern, 3: color)

5:Q1Feat Questioned feature 1. A number between 1 and 3. (1: shape, 2: pattern, 3: color).

Example: If you utter "blue" (feature: color), then you can learn something about shape and texture preferences.

6:Q2Feat Questioned feature 2. A number between 1 and 3. (1: shape, 2: pattern, 3: color).

Example: If you utter "blue" (feature: color), then you can learn something about shape and texture preferences.

7:Q1AnswerV1, V2, V3 The columns 7-9 contain the participants' slider values for the first questioned feature.

10:Q2AnswerV1, V2, C3 The columns 10-12 contain the participants' slider values for the second questioned feature.

Details

This function uses `RSAModelKLDiv3params_simpleRSA4TrialsIterative_dep`.

Value

Minimized Kullback-Leibler divergence and the optimal parameter values.

`LL1_1_Iterative_dep_notObey0.1`

Cost function for one parameter optimization (iterative setting, dependent on trial order). Optimizing softness. Non-obedience fixed at 0.1.

Description

Simple RSA

1 parameter optimization; The softness parameter is optimized.

The non-obedience parameter is fixed.

Usage

```
LL1_1_Iterative_dep_notObey0.1(params, data)
```

Arguments

`params`	One value vector, which specifies one of two parameters to be optimized: 1. softPrefValue is optimized, i.e. The strength of "preferring one entity over others". (The larger the value the higher the tendency towards uniform liking) 2. non-obedience fixed at 0.1, i.e. The extent to which the instruction of the speaker is obeyed by the listener. (0 = full obedience, infinity = full instruction ignorance)

`data` A Matrix with data rows.

column structure:

[1:OC1,OC2,OC3,4:UUFeat, 5:Q1Feat,6:Q2Feat]

[7:Q1AnswerV1,V2,V3, 10:Q2AnswerV1,V2,V3]

1:OC1 Object 1. A value between 1 and 27.

2:OC2 Object 2. A value between 1 and 27.

3:OC3 Object 3. A value between 1 and 27.

4:UUFeat Uttered feature. A number between 1 and 3. (1: shape, 2: pattern, 3: color)

5:Q1Feat Questioned feature 1. A number between 1 and 3. (1: shape, 2: pattern, 3: color).

Example: If you utter "blue" (feature: color), then you can learn something about shape and texture preferences.

6:Q2Feat Questioned feature 2. A number between 1 and 3. (1: shape, 2: pattern, 3: color).

Example: If you utter "blue" (feature: color), then you can learn something about shape and texture preferences.

7:Q1AnswerV1, V2, V3 The columns 7-9 contain the participants' slider values for the first questioned feature.

10:Q2AnswerV1, V2, C3 The columns 10-12 contain the participants' slider values for the second questioned feature.

Details

This function uses `RSAModelKLDiv3params_simpleRSA4TrialsIterative_dep`.

Value

Minimized Kullback-Leibler divergence and the optimal parameter values.

`LL1_1_Iterative_pr_notObey0_pr0.5`

Cost function for one parameter optimization (iterative setting, independent of trial order). Optimizing softness. non-obedience is fixed at 0. prior rate is fixed at 0.5.

Description

Simple RSA

1 parameter optimization; The softness parameter is optimized. (1st)

The non-obedience and prior rate parameter are fixed.

Usage

```
LL1_1_Iterative_pr_notObey0_pr0.5(params, data)
```

Arguments

`params` One value vector, which specifies one of three parameters to be optimized:

1. softPrefValue is optimized, i.e. The strength of "preferring one entity over others". (The larger the value the higher the tendency towards uniform liking)
2. non-obedience is fixed at 0, i.e. The extent to which the instruction of the speaker is obeyed by the listener. (0 = full obedience, infinity = full instruction ignorance)
3. priorRate is fixed to 0.5. This parameter specifies how much the prior information is weighed into the decision .

`data` A Matrix with data rows.

column structure:

[1:OC1,OC2,OC3,4:UUFeat, 5:Q1Feat,6:Q2Feat]

[7:Q1AnswerV1,V2,V3, 10:Q2AnswerV1,V2,V3]

1:OC1 Object 1. A value between 1 and 27.

2:OC2 Object 2. A value between 1 and 27.

3:OC3 Object 3. A value between 1 and 27.

4:UUFeat Uttered feature. A number between 1 and 3. (1: shape, 2: pattern, 3: color)

5:Q1Feat Questioned feature 1. A number between 1 and 3. (1: shape, 2: pattern, 3: color).

Example: If you utter "blue" (feature: color), then you can learn something about shape and texture preferences.

6:Q2Feat Questioned feature 2. A number between 1 and 3. (1: shape, 2: pattern, 3: color).

Example: If you utter "blue" (feature: color), then you can learn something about shape and texture preferences.

7:Q1AnswerV1, V2, V3 The columns 7-9 contain the participants' slider values for the first questioned feature.

10:Q2AnswerV1, V2, C3 The columns 10-12 contain the participants' slider values for the second questioned feature.

Details

This function uses `RSAModelKLDiv3params_simpleRSA4TrialsIterative_pr`.

Value

Minimized Kullback-Leibler divergence and the optimal parameter values.

`LL1_2_Iterative_dep_pref0`

Cost function for one parameter optimization (iterative setting, dependent on trial order). Optimizing non-obedience. Softness is fixed at 0.

Description

Simple RSA

1 parameter optimization; The softness parameter is optimized.

The non-obedience parameter is fixed.

Usage

```
LL1_2_Iterative_dep_pref0(params, data)
```

Arguments

`params`	One value vector, which specifies one of two parameters to be optimized: 1. softPrefValue is fixed to 0, i.e. The strength of "preferring one entity over others". (The larger the value the higher the tendency towards uniform liking) 2. non-obedience is optimized i.e. The extent to which the instruction of the speaker is obeyed by the listener. (0 = full obedience, infinity = full instruction ignorance)
`data`	A Matrix with data rows. column structure: [1:OC1,OC2,OC3,4:UUFeat, 5:Q1Feat,6:Q2Feat] [7:Q1AnswerV1,V2,V3, 10:Q2AnswerV1,V2,V3] **1:OC1** Object 1. A value between 1 and 27. **2:OC2** Object 2. A value between 1 and 27. **3:OC3** Object 3. A value between 1 and 27. **4:UUFeat** Uttered feature. A number between 1 and 3. (1: shape, 2: pattern, 3: color) **5:Q1Feat** Questioned feature 1. A number between 1 and 3. (1: shape, 2: pattern, 3: color). Example: If you utter "blue" (feature: color), then you can learn something about shape and texture preferences. **6:Q2Feat** Questioned feature 2. A number between 1 and 3. (1: shape, 2: pattern, 3: color). Example: If you utter "blue" (feature: color), then you can learn something about shape and texture preferences. **7:Q1AnswerV1, V2, V3** The columns 7-9 contain the participants' slider values for the first questioned feature. **10:Q2AnswerV1, V2, C3** The columns 10-12 contain the participants' slider values for the second questioned feature.

Details

This function uses `RSAModelKLDiv3params_simpleRSA4TrialsIterative_dep`.

Value

Minimized Kullback-Leibler divergence and the optimal parameter values.

`LL1_2_Iterative_pr_pref0_pr0.5`

Cost function for one parameter optimization (iterative setting, independent of trial order). Optimizing non-obedience. softness is fixed at 0. prior rate is fixed at 0.5.

Description

Simple RSA

1 parameter optimization; The non-obedience parameter is optimized. (2nd)

The non-obedience and prior rate parameter are fixed.

Usage

```
LL1_2_Iterative_pr_pref0_pr0.5(params, data)
```

Arguments

`params`	One value vector, which specifies one of three parameters to be optimized: 1. softPrefValue is fixed at 0, i.e. The strength of "preferring one entity over others". (The larger the value the higher the tendency towards uniform liking) 2. non-obedience is optimized, i.e. The extent to which the instruction of the speaker is obeyed by the listener. (0 = full obedience, infinity = full instruction ignorance) 3. priorRate is fixed to 0.5. This parameter specifies how much the prior information is weighed into the decision is weighed into the decision of the speaker regarding the feature preferences of the listener.
`data`	A Matrix with data rows. column structure: [1:OC1,OC2,OC3,4:UUFeat, 5:Q1Feat,6:Q2Feat] [7:Q1AnswerV1,V2,V3, 10:Q2AnswerV1,V2,V3] **1:OC1** Object 1. A value between 1 and 27. **2:OC2** Object 2. A value between 1 and 27. **3:OC3** Object 3. A value between 1 and 27. **4:UUFeat** Uttered feature. A number between 1 and 3. (1: shape, 2: pattern, 3: color) **5:Q1Feat** Questioned feature 1. A number between 1 and 3. (1: shape, 2: pattern, 3: color). Example: If you utter "blue" (feature: color), then you can learn something about shape and texture preferences. **6:Q2Feat** Questioned feature 2. A number between 1 and 3. (1: shape, 2: pattern, 3: color). Example: If you utter "blue" (feature: color), then you can learn something about shape and texture preferences. **7:Q1AnswerV1, V2, V3** The columns 7-9 contain the participants' slider values for the first questioned feature. **10:Q2AnswerV1, V2, C3** The columns 10-12 contain the participants' slider values for the second questioned feature.

Details

This function uses RSAModelKLDiv3params_simpleRSA4TrialsIterative_pr.

Value

Minimized Kullback-Leibler divergence and the optimal parameter values.

LL1_3_Iterative_pr_pref0_notObey0

Cost function for one parameter optimization (iterative setting, independent of trial order). Optimizing the prior rate. softness and non-obedience are fixed at 0.

Description

Simple RSA (iterative, independent of trial order)

1 parameter optimization; The prior rate parameter is optimized.

The softness and non-obedience parameters are fixed.

Usage

```
LL1_3_Iterative_pr_pref0_notObey0(params, data)
```

Arguments

params	One value vector, which specifies one of three parameters to be optimized: 1. softPrefValue is fixed at 0, i.e. The strength of "preferring one entity over others". (The larger the value the higher the tendency towards uniform liking) 2. non-obedience is fixed at 0 i.e. The extent to which the instruction of the speaker is obeyed by the listener. (0 = full obedience, infinity = full instruction ignorance) 3. priorRate is optimized. This parameter specifies how much the prior information is weighed into the decision.
data	A Matrix with data rows. column structure: [1:OC1,OC2,OC3,4:UUFeat, 5:Q1Feat,6:Q2Feat] [7:Q1AnswerV1,V2,V3, 10:Q2AnswerV1,V2,V3] **1:OC1** Object 1. A value between 1 and 27. **2:OC2** Object 2. A value between 1 and 27. **3:OC3** Object 3. A value between 1 and 27. **4:UUFeat** Uttered feature. A number between 1 and 3. (1: shape, 2: pattern, 3: color) **5:Q1Feat** Questioned feature 1. A number between 1 and 3. (1: shape, 2: pattern, 3: color). Example: If you utter "blue" (feature: color), then you can learn something about shape and texture preferences.

6:Q2Feat Questioned feature 2. A number between 1 and 3. (1: shape, 2: pattern, 3: color).

Example: If you utter "blue" (feature: color), then you can learn something about shape and texture preferences.

7:Q1AnswerV1, V2, V3 The columns 7-9 contain the participants' slider values for the first questioned feature.

10:Q2AnswerV1, V2, C3 The columns 10-12 contain the participants' slider values for the second questioned feature.

Details

This function uses `RSAModelKLDiv3params_simpleRSA4TrialsIterative_pr`.

Value

Minimized Kullback-Leibler divergence and the optimal parameter values.

`LL2_12_Iterative_dep` *Cost function for one parameter optimization (iterative setting, dependent on trial order). Optimizing softness and non-obedience.*

Description

Simple RSA

2 parameter optimization; The softness and non-obedience parameters are optimized.

The non-obedience parameter is fixed.

Usage

```
LL2_12_Iterative_dep(params, data)
```

Arguments

`params`	One value vector, which specifies one of two parameters to be optimized: 1. softPrefValue is optimized, i.e. The strength of "preferring one entity over others". (The larger the value the higher the tendency towards uniform liking) 2. non-obedience is optimized, i.e. The extent to which the instruction of the speaker is obeyed by the listener. (0 = full obedience, infinity = full instruction ignorance)
`data`	A Matrix with data rows. column structure: [1:OC1,OC2,OC3,4:UUFeat, 5:Q1Feat,6:Q2Feat] [7:Q1AnswerV1,V2,V3, 10:Q2AnswerV1,V2,V3] **1:OC1** Object 1. A value between 1 and 27. **2:OC2** Object 2. A value between 1 and 27. **3:OC3** Object 3. A value between 1 and 27. **4:UUFeat** Uttered feature. A number between 1 and 3. (1: shape, 2: pattern, 3: color)

5:Q1Feat Questioned feature 1. A number between 1 and 3. (1: shape, 2: pattern, 3: color).

Example: If you utter "blue" (feature: color), then you can learn something about shape and texture preferences.

6:Q2Feat Questioned feature 2. A number between 1 and 3. (1: shape, 2: pattern, 3: color).

Example: If you utter "blue" (feature: color), then you can learn something about shape and texture preferences.

7:Q1AnswerV1, V2, V3 The columns 7-9 contain the participants' slider values for the first questioned feature.

10:Q2AnswerV1, V2, C3 The columns 10-12 contain the participants' slider values for the second questioned feature.

Details

This function uses `RSAModelKLDiv3params_simpleRSA4TrialsIterative_dep`.

Value

Minimized Kullback-Leibler divergence and the optimal parameter values.

`LL2_13_Iterative_pr_notObey0`

Cost function for one parameter optimization (iterative setting, independent of trial order). Optimizing the softness and prior rate. non-obedience is fixed at 0.

Description

Simple RSA (iterative, independent of trial order)

2 parameter optimization; The softness and prior rate parameter are optimized (1st and 3rd)

The softness and prior rate parameters are fixed.

Usage

```
LL2_13_Iterative_pr_notObey0(params, data)
```

Arguments

`params` One value vector, which specifies one of three parameters to be optimized:

1. softPrefValue is optimized, i.e. The strength of "preferring one entity over others". (The larger the value the higher the tendency towards uniform liking)
2. non-obedience is fixed at 0 i.e. The extent to which the instruction of the speaker is obeyed by the listener. (0 = full obedience, infinity = full instruction ignorance)
3. priorRate is optimized. This parameter specifies how much the prior information is weighed into the decision.

`data` A Matrix with data rows.

column structure:

[1:OC1,OC2,OC3,4:UUFeat, 5:Q1Feat,6:Q2Feat]

[7:Q1AnswerV1,V2,V3, 10:Q2AnswerV1,V2,V3]

1:OC1 Object 1. A value between 1 and 27.

2:OC2 Object 2. A value between 1 and 27.

3:OC3 Object 3. A value between 1 and 27.

4:UUFeat Uttered feature. A number between 1 and 3. (1: shape, 2: pattern, 3: color)

5:Q1Feat Questioned feature 1. A number between 1 and 3. (1: shape, 2: pattern, 3: color).

Example: If you utter "blue" (feature: color), then you can learn something about shape and texture preferences.

6:Q2Feat Questioned feature 2. A number between 1 and 3. (1: shape, 2: pattern, 3: color).

Example: If you utter "blue" (feature: color), then you can learn something about shape and texture preferences.

7:Q1AnswerV1, V2, V3 The columns 7-9 contain the participants' slider values for the first questioned feature.

10:Q2AnswerV1, V2, C3 The columns 10-12 contain the participants' slider values for the second questioned feature.

Details

This function uses `RSAModelKLDiv3params_simpleRSA4TrialsIterative_pr`.

Value

Minimized Kullback-Leibler divergence and the optimal parameter values.

`LL2_13_Iterative_pr_notObey0.1`

Cost function for one parameter optimization (iterative setting, independent of trial order). Optimizing the softness and prior rate. non-obedience is fixed at 0.1.

Description

Simple RSA (iterative, independent of trial order)

2 parameter optimization; The softness and prior rate parameter are optimized (1st and 3rd)

The softness and prior rate parameters are fixed.

Usage

```
LL2_13_Iterative_pr_notObey0.1(params, data)
```

Arguments

`params` One value vector, which specifies one of three parameters to be optimized:

1. softPrefValue is optimized, i.e. The strength of "preferring one entity over others". (The larger the value the higher the tendency towards uniform liking)
2. non-obedience is fixed at 0 i.e. The extent to which the instruction of the speaker is obeyed by the listener. (0 = full obedience, infinity = full instruction ignorance)
3. priorRate is optimized. This parameter specifies how much the prior information is weighed into the decision.

`data` A Matrix with data rows.

column structure:

[1:OC1,OC2,OC3,4:UUFeat, 5:Q1Feat,6:Q2Feat]

[7:Q1AnswerV1,V2,V3, 10:Q2AnswerV1,V2,V3]

1:OC1 Object 1. A value between 1 and 27.

2:OC2 Object 2. A value between 1 and 27.

3:OC3 Object 3. A value between 1 and 27.

4:UUFeat Uttered feature. A number between 1 and 3. (1: shape, 2: pattern, 3: color)

5:Q1Feat Questioned feature 1. A number between 1 and 3. (1: shape, 2: pattern, 3: color).

Example: If you utter "blue" (feature: color), then you can learn something about shape and texture preferences.

6:Q2Feat Questioned feature 2. A number between 1 and 3. (1: shape, 2: pattern, 3: color).

Example: If you utter "blue" (feature: color), then you can learn something about shape and texture preferences.

7:Q1AnswerV1, V2, V3 The columns 7-9 contain the participants' slider values for the first questioned feature.

10:Q2AnswerV1, V2, C3 The columns 10-12 contain the participants' slider values for the second questioned feature.

Details

This function uses `RSAModelKLDiv3params_simpleRSA4TrialsIterative_pr`.

Value

Minimized Kullback-Leibler divergence and the optimal parameter values.

`LL2_23_Iterative_pr_pref0`

Cost function for one parameter optimization (iterative setting, independent of trial order). `posterior = (1 - prior rate) x evidence + (prior rate) x prior` *Optimizing the non-obedience and prior rate. softness is fixed at 0.*

Description

Simple RSA (iterative, independent of trial order)

2 parameter optimization; The softness and prior rate parameter are optimized (1st and 3rd)

The softness and prior rate parameters are fixed.

Usage

```
LL2_23_Iterative_pr_pref0(params, data)
```

Arguments

`params`	One value vector, which specifies one of three parameters to be optimized: 1. softPrefValue is fixed at 0, i.e. The strength of "preferring one entity over others". (The larger the value the higher the tendency towards uniform liking) 2. non-obedience is optimized, i.e. The extent to which the instruction of the speaker is obeyed by the listener. (0 = full obedience, infinity = full instruction ignorance) 3. priorRate is optimized. This parameter specifies how much the prior information is weighed into the decision of the speaker regarding the feature preferences of the listener.
`data`	A Matrix with data rows. column structure: [1:OC1,OC2,OC3,4:UUFeat, 5:Q1Feat,6:Q2Feat] [7:Q1AnswerV1,V2,V3, 10:Q2AnswerV1,V2,V3] **1:OC1** Object 1. A value between 1 and 27. **2:OC2** Object 2. A value between 1 and 27. **3:OC3** Object 3. A value between 1 and 27. **4:UUFeat** Uttered feature. A number between 1 and 3. (1: shape, 2: pattern, 3: color) **5:Q1Feat** Questioned feature 1. A number between 1 and 3. (1: shape, 2: pattern, 3: color). Example: If you utter "blue" (feature: color), then you can learn something about shape and texture preferences. **6:Q2Feat** Questioned feature 2. A number between 1 and 3. (1: shape, 2: pattern, 3: color). Example: If you utter "blue" (feature: color), then you can learn something about shape and texture preferences. **7:Q1AnswerV1, V2, V3** The columns 7-9 contain the participants' slider values for the first questioned feature. **10:Q2AnswerV1, V2, C3** The columns 10-12 contain the participants' slider values for the second questioned feature.

Details

This function uses `RSAModelKLDiv3params_simpleRSA4TrialsIterative_pr`.

Value

Minimized Kullback-Leibler divergence and the optimal parameter values.

`RSAModelKLDiv3params_simpleRSA4TrialsIterative_dep`
Simple RSA model Kullback-Leibler divergence determination (iterative setting, dependent on trial order)

Description

Simple RSA (iterative, dependent on trial order)

The function calculates the optimal parameter values of the free parameters by estimating the log-likelihood of the RSA model given model parameters and data. It also determines the actual RSA model Kullback-Leibler divergence.

2 parameter optimization considering only the available feature values present in the scene, i.e. feature values of shape, texture and color. This function is used in the iterative dependent on the trial scenario.

Usage

```
RSAModelKLDiv3params_simpleRSA4TrialsIterative_dep(data, par1, par2)
```

Arguments

`data`	A matrix with data rows. column structure: `[1:OC1,OC2,OC3,4:numUttOptions,7-X:TurkerSliderValues]` **1:OC1** Object 1. A value between 1 and 27. **2:OC2** Object 2. A value between 1 and 27. **3:OC3** Object 3. A value between 1 and 27. **4:numUttOptions** The number of valid utterances in the scene. **7-X:TurkerSliderValues** These columns contain the participants' slider values.
`par1`	**softness parameter** The strength of "preferring one entity over others". (The larger the value the higher the tendency towards uniform liking)
`par2`	**non-obedience parameter** The extent to which the instruction of the speaker is obeyed by the listener. (0 = full obedience, infinity = full instruction ignorance).

Details

This function is used in `LL1_1_Iterative_dep_notObey0`,

`LL1_1_Iterative_dep_notObey0.1`,

`LL1_2_Iterative_dep_pref0`,

`LL2_12_Iterative_dep`.

Value

Minimized Kullback-Leibler divergence and the optimal parameter values.

Minimized Kullback-Leibler divergence and the optimal parameters.

`RSAModelKLDiv3params_simpleRSA4TrialsIterative_pr` *Simple RSA model Kullback-Leibler divergence determination (iterative setting, independent of trial order)*

Description

Simple RSA (iterative, independent of trial order (prior rate))

The function calculates the optimal parameter values of the free parameters by estimating the log-likelihood of the RSA model given model parameters and data. It also determines the actual RSA model Kullback-Leibler divergence.

2 parameter optimization considering only the available feature values present in the scene, i.e. feature values of shape, texture and color. This function is used in the iterative dependent on the trial scenario.

Usage

```
RSAModelKLDiv3params_simpleRSA4TrialsIterative_pr(data, par1, par2, par3)
```

Arguments

`data`	A matrix with data rows. column structure: `[1:OC1,OC2,OC3,4:numUttOptions,7-X:TurkerSliderValues]` **1:OC1** Object 1. A value between 1 and 27. **2:OC2** Object 2. A value between 1 and 27. **3:OC3** Object 3. A value between 1 and 27. **4:numUttOptions** The number of valid utterances in the scene. **7-X:TurkerSliderValues** These columns contain the participants' slider values.
`par1`	**softness parameter** The strength of "preferring one entity over others". (The larger the value the higher the tendency towards uniform liking)
`par2`	**non-obedience parameter** The extent to which the instruction of the speaker is obeyed by the listener. (0 = full obedience, infinity = full instruction ignorance).
`par3`	**prior rate parameter** This parameter specifies how much the prior information is weighed into the speaker's decision is weighed into the decision of the speaker regarding the feature preferences of the listener.

Details

This function is used in `LL1_1_Iterative_pr_notObey0_pr0.5`,

`LL1_2_Iterative_pr_pref0_pr0.5`,

`LL1_3_Iterative_pr_pref0_notObey0`,

`LL2_13_Iterative_pr_notObey0`,

`LL2_13_Iterative_pr_notObey0.1`,

`LL2_23_Iterative_pr_pref0`.

Value

Minimized Kullback-Leibler divergence and the optimal parameter values.

Minimized Kullback-Leibler divergence and the optimal parameters.

`simplePragmaticSpeakerWithPrefPriorAll_depOnOrder`
Simple pragmatic speaker with all prior preferences iterative function
Iterative function dependent on trial order (prior rate)

Description

Simple-RSA (iterative, dependent on trial order)

The simple pragmatic speaker considers all "imaginable" (i.e. implemented) preference distributions over objects of the listener.

Starting with a prior assumption over the possible listener's preferences. It then infers the posterior over these preferences given the listener makes a particular object choice. P(listener's feature value preferences | utterance, object choice by the listener, prior over listener's feature value preferences).

This function takes the evidence from the current trial in consideration and also the prior from the trials before and computes the posterior over the feature preferences of the listener. `posterior = (1 -prior rate) x evidence + (prior rate) x prior`.

Usage

```
simplePragmaticSpeakerWithPrefPriorAll_depOnOrder(
  utterance,
  obj,
  preferencesPriorAll,
  validUtterances,
  currentObjects,
  uttToObjProbs,
  objectPreferenceSoftPriors
)
```

Arguments

`utterance`	The uttered word by the speaker that the listener hears. An index referring to one of the values in the vector validUtterances.
`obj`	The object chosen by the listener. A value referring to the index 1,2 or 3.
`preferencesPriorAll`	A vector of length 9. Probability mass over all feature values. Gives a prior preferences distribution over all (nine) feature values. `preferencesPriorAll <-rep(1/9,9)`
`validUtterances`	A vector of utterances that correspond to all feature values present in the current objects in the scene. For example, it only makes sense to utter *"red"* in a scene if there are *red* objects present.

`currentObjects` Vector of three values in {1,...,27} specifying the target and the other two objects.

The target is the first object in the vector (index = 1).

`uttToObjProbs` A matrix. The rows map each possible utterance that corresponds to each present feature value of the current objects. The columns represent the three objects in the scene.

This reflects the obedience-parameter and which objects match the respective utterance. The matrix shows the probability that a certain object is chosen following a certain utterance, that is valid in the scene. The number of rows of the matrix match the length of the validUtterances vector.

`objectPreferenceSoftPriors`
A list of preference priors for all valid utterances based on the object in the scene.

The list has as many rows as the length of the validUtterances vector + 1.

Each row in the list contains a vector of length 3, as there are three objects in the scene.

The extra row is for the case of no feature preferences whatsoever, i.e. uniform prior over all three objects in the scene.

`weights` A vector of length 4 including the weight by which the prior is weighed.

weights <- c(0.3, 0.4, 0.5, 0.6)

`trial` A vector of length 4 including the number of the current trial.

trial <- c(1,2,3,4)

Details

This is function is the second of two functions that are used in the iterative setting using the prior rate parameter. The first one is: `simplePragmaticSpeakerWithPrefPriorAll_indepOfOrder_pr`.

Value

A vector of length 9. It contains the normalized probability over preferences (priors).

Examples

```
simplePragmaticSpeakerWithPrefPriorAll_depOnOrder(utterance, obj, preferencesPriorAll,
validUtterances, currentObjects,
uttToObjProbs, objectPreferenceSoftPriors)

output:
[1] 0.5333333 0.1333333 0.3333333 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
[9] 0.0000000
```

`simplePragmaticSpeakerWithPrefPriorAll_indepOfOrder_pr`
Simple pragmatic speaker with all prior preferences iterative function Iterative function independent of trial order (prior rate). `posterior = (1 - prior rate) x evidence + (prior rate) x prior`.

Description

Simple-RSA (iterative, independent on trial order) The simple pragmatic speaker considers all "imaginable" (i.e. implemented) preference distributions over objects of the listener.

Starting with a prior assumption over the possible listener's preferences. It then infers the posterior over these preferences given the listener makes a particular object choice. P(listener's feature value preferences | utterance, object choice by the listener, prior over listener's feature value preferences).

This function takes the evidence from the current trial in consideration and also the prior from the trials before and computes the posterior over the feature preferences of the listener. `posterior = (1 -prior rate) x evidence + (prior rate) x prior`.

Usage

```
simplePragmaticSpeakerWithPrefPriorAll_indepOfOrder_pr(
  utterance,
  obj,
  preferencesPriorAll,
  validUtterances,
  currentObjects,
  uttToObjProbs,
  objectPreferenceSoftPriors,
  priorRate
)
```

Arguments

`utterance`	The uttered word by the speaker that the listener hears. An index referring to one of the values in the vector validUtterances.
`obj`	The object chosen by the listener. A value referring to the index 1,2 or 3.
`preferencesPriorAll`	A vector of length 9. Probability mass over all feature values. Gives a prior preferences distribution over all (nine) feature values. `preferencesPriorAll <-rep(1/9,9)`
`validUtterances`	A vector of utterances that correspond to all feature values present in the current objects in the scene. For example, it only makes sense to utter *"red"* in a scene if there are *red* objects present.
`currentObjects`	Vector of three values in `{1,...,27}` specifying the target and the other two objects. The target is the first object in the vector `(index = 1)`.
`uttToObjProbs`	A matrix. The rows map each possible utterance that corresponds to each present feature value of the current objects. The columns represent the three objects in the scene. This reflects the obedience-parameter and which objects match the respective utterance. The matrix shows the probability that a certain object is chosen following a certain utterance, that is valid in the scene. The number of rows of the matrix match the length of the validUtterances vector.

`objectPreferenceSoftPriors`
: A list of preference priors for all valid utterances based on the object in the scene.
The list has as many rows as the length of the validUtterances vector + 1.
Each row in the list contains a vector of length 3, as there are three objects in the scene.
The extra row is for the case of no feature preferences whatsoever, i.e. uniform prior over all three objects in the scene.

`priorRate`
: A parameter specifying how much prior information is weighed into the decision.

Details

This is function is the first of two functions that are used in the iterative setting using the prior rate parameter. The second one is: `simplePragmaticSpeakerWithPrefPriorAll_indepOfOrder_pr`.

Value

A vector of length 9. It contains the normalized probability over preferences (priors).

Examples

```
simplePragmaticSpeakerWithPrefPriorAll_indepOfOrder_pr(utterance, obj,
preferencesPriorAll, validUtterances, currentObjects, uttToObjProbs, objectPreferenceSoftPriors, priorRate)

output:
[1] 0.5333333 0.1333333 0.3333333 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
[9] 0.0000000
```

Index