Package 'priorinference'

February 17, 2021

Type Package

Title Prior Inference Package

Version 0.1.0

Author Martin V. Butz, Hania Gaber Elkersh, Asya Achimova, Gregory Scontras

Maintainer Hania Gaber Elkersh <hania.elkersh@student.uni-tuebingen.de>

Description This package contains all the functions of the common code of the prior inference and utterance choice models.
This package was designed to analyze and model data for Experiments 1 and 2 in the paper Learning about others: Modeling social inference through ambiguity resolution (Achimova et al. 2020).

URL http://www.socsci.uci.edu/~gscontra/experiments/prior_inference/inferring-preferences/prior-inference.html

http://www.socsci.uci.edu/~gscontra/experiments/prior_inference/utterance-choice/prior-inference.html

License MIT License

Encoding UTF-8

LazyData true

Suggests testthat,
knitr,
rmarkdown

RoxygenNote 7.1.1

Imports

VignetteBuilder knitr

R topics documented:

`allFeatureTypesNew` *All feature types*

Description

A vector of all object features: ('shape','pattern','color')

Usage

```
allFeatureTypesNew
```

Format

An object of class `character` of length 3.

allObjects *All objects*

Description

The matrix contains `3^3` types of objects.

It essentially specifies the 3 feature expressions for each object thus, the matrix maps objects to matching utterances. The matrix contains strings of all possible feature values of the objects.

The index mapping of the strings are contained in the matrix `allObjectsToUtterancesMappings`.

Usage

```
allObjects
```

Format

An object of class `matrix` (inherits from `array`) with 27 rows and 3 columns.

Examples

```
## Not run:
output:
     shape   pattern  color
[1,] cloud    solid   blue
[2,] circle   solid   blue
[3,] square   solid   blue
[4,] cloud   striped  blue
[5,] circle  striped  blue
[6,] square  striped  blue
[7,] cloud   dotted   blue
[8,] circle  dotted   blue
[9,] square  dotted   blue
[10,] cloud    solid    red
[11,] circle   solid    red
[12,] square   solid    red
[13,] cloud   striped   red
[14,] circle  striped   red
[15,] square  striped   red
[16,] cloud   dotted    red
[17,] circle  dotted    red
[18,] square  dotted    red
[19,] cloud    solid    green
[20,] circle   solid    green
[21,] square   solid    green
[22,] cloud   striped   green
[23,] circle  striped   green
[24,] square  striped   green
[25,] cloud   dotted    green
[26,] circle  dotted    green
[27,] square  dotted    green

## End(Not run)
```

`allObjectsToUtterancesMappings`
All objects to utterances mappings

Description

The matrix contains `3^3` types of objects.

It essentially specifies the 3 feature expressions for each object thus, the matrix maps objects to matching utterances. The matrix contains the index mapping of all possible feature values:

cloud = 1, circle = 2, square = 3

solid = 4, striped = 5, dotted = 6

blue = 7, red = 8, green = 9

The strings of the index mapping are contained in the matrix `allObjects`.

Usage

```
allObjectsToUtterancesMappings
```

Format

An object of class `matrix` (inherits from `array`) with 27 rows and 3 columns.

Examples

```
## Not run:
output:

      [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    4    7
[3,]    3    4    7
[4,]    1    5    7
[5,]    2    5    7
[6,]    3    5    7
[7,]    1    6    7
[8,]    2    6    7
[9,]    3    6    7
[10,]    1    4    8
[11,]    2    4    8
[12,]    3    4    8
[13,]    1    5    8
[14,]    2    5    8
[15,]    3    5    8
[16,]    1    6    8
[17,]    2    6    8
[18,]    3    6    8
[19,]    1    4    9
[20,]    2    4    9
[21,]    3    4    9
[22,]    1    5    9
[23,]    2    5    9
[24,]    3    5    9
```

```
[25,]    1    6    9
[26,]    2    6    9
[27,]    3    6    9

## End(Not run)
```

`allUtterances` *All utterances*

Description

A vector that contains all feature values:

('cloud', 'circle', 'square', 'solid', 'striped', 'dotted', 'blue', 'red', 'green')

Usage

```
allUtterances
```

Format

An object of class `character` of length 9.

Details

The difference between this matrix and the `allUtterancesNew` is that the 'dotted' pattern is called *polka-dotted*.

`allUtterancesNew` *All utterances (new)*

Description

A vector that contains all feature values:

('cloud', 'circle', 'square', 'solid', 'striped', 'polka-dotted', 'blue', 'red', 'green')

Usage

```
allUtterancesNew
```

Format

An object of class `character` of length 9.

Details

The difference between this vector and the `allUtterances` is that the 'polka-dotted' pattern is called *dotted*.

`bestInfGainUtterance` *Best information gain utterances*

Description

Full-RSA

The ultimate function that determines the utterance preferences of a rather "informed", "pragmatic" speaker considering all possible scenarios. That is, hypothetically, all utterances are considered. Additionally, the resulting inferred listener's object preferences are computed assuming the listener picks a certain object and has certain object preferences.

U(utt | listener's object preference priors) is computed. The utility is determined as the information gain between prior and posterior of the determined listener's object preferences.

Usage

```
bestInfGainUtterance(
  preferencesPrior,
  validUtterances,
  currentObjects,
  uttToObjProbs,
  objectPreferenceSoftPriors,
  alpha = 1,
  klValueFactor = 1
)
```

Arguments

`preferencesPrior`

A vector of the length the validUtterances vector + 1.

It constructed as such:

`preferencesPrior <-rep(1/(length(validUtterances)+1),length(validUtterances)+1).`

The vector contains the probability mass over all feature values present in the scenario plus a "no preference" case.

Gives a prior preferences distribution over the feature values in the scene.

`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` 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)`.

`uttToObjProbs` A matrix.

The rows map each possible utterance that corresponds to each present feature value of the current objects. The number of rows of the matrix match the length of the validUtterances vector.

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.

`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 preferences whatsoever, i.e. uniform prior over all three objects in the scene.

`alpha`
: A parameter between 0 and 1. (Here it's set to = 1)

  Exponential scaling of the speaker choosing the utterance that maximizes the chance of the listener getting the target object right.

`klValueFactor`
: A parameter that can be negative, 0 or positive (Here it is set to = 1):

  zero Don't care about learning about the feature preferences of the listener

  positive Care about learning about the feature preferences of the listener

  negative Trying to pick non-ambiguous utterances

Value

A vector containing the normalized probability over utterances given the listener's object preference priors.

The utterance with the highest probability is the one that maximizes the information gain for the speaker.

The vector has the same length as the validUtterances vector.

Examples

```
currentObjects <- c(1,2,3)
allObjects[currentObjects,]

output:

      shape    pattern color
[1,] "cloud"  "solid" "blue"
[2,] "circle" "solid" "blue"
[3,] "square" "solid" "blue"


bestInfGainUtterance(preferencesPrior, validUtterances,

currentObjects,uttToObjProbs, objectPreferenceSoftPriors,

alpha=1, klValueFactor=1)

output:
[1] 0.0 0.0 0.0 0.5

Since the all the objects present in the scene are solid and blue,
uttering solid or blue, would be optimal to learn something
about the shape preferences of the listener.
This means the speaker would have the best information gain.
```

`determineSpeakerPostListPrefs` *Determine speaker's inference of the posterior listener preferences*

Description

Full-RSA

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

Usage

```
determineSpeakerPostListPrefs(
  currentObjects,
  featureUtt,
  softPrefValue,
  notObeyInst,
  alpha
)
```

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"*.
`alpha`	A parameter value between 0 and 1. Exponential scaling of the speaker choosing the utterance that maximizes the chance of the listener getting the target object right.

Value

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

`determineSpeakerPostListPrefsSimpleRSA`
Determine speaker's inference of the posterior listener preferences

Description

Simple RSA

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

Usage

```
determineSpeakerPostListPrefsSimpleRSA(
  currentObjects,
  featureUtt,
  softPrefValue,
  notObeyInst
)
```

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"*.

Value

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

`determineSpeakerPostListPrefsSimpleRSAWithPriorPref` *Determine speaker's inference of the posterior listener preferences (iterative setting)*

Description

Simple RSA

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(
  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.

Details

This is the iterative version of the function `determineSpeakerPostListPrefsSimpleRSA`

Value

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

Examples

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

output:
[1] 0.33 0.00 0.00 0.11 0.11 0.11 0.11 0.11 0.11
```

---

determineUttToObjectProbs

Determine utterances to object probabilities

Description

This function determines maps the valid utterances to the corresponding objects in the scene.

It then determines the probability of choosing an object based on the valid utterances in the scene.

Usage

```
determineUttToObjectProbs(
  validUtterances,
  currentObjects,
  mapObjToUtt,
  notObeyInst
)
```

Arguments

`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`	A vector of three values in range `{1,...,27}` specifying the target and the other two objects in the scene. The target is the first object in the vector `(index = 1)`.
`mapObjToUtt`	A 3x3 matrix. Values in the matrix are in range `{1-9}`. **rows:** The current objects in the scene **columns:** Features of the 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"*.

Details

This function is only used in X9

Value

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.

Examples

```
determineUttToObjectProbs(validUtterances, currentObjects, mappedObjToUtt, notObeyInst)

output:
     [,1] [,2] [,3]
[1,] 1.00 0.00 0.00
[2,] 0.00 1.00 0.00
[3,] 0.00 0.00 1.00
[4,] 0.33 0.33 0.33
[5,] 0.33 0.33 0.33

Example:

To see which objects are present in the scene run:
allObjects[currentObjects,]

output:

      shape   pattern  color
[1,] "cloud"  "solid" "blue"
[2,] "circle" "solid" "blue"
[3,] "square" "solid" "blue"


The columns correspond to the three objects in the scene:
A solid blue cloud, a solid blue circle and a solid blue square.
If the utterance is circle the probability of choosing circle is 1,
while the probability for the cloud and the square is 0.
If the utterance is blue the probability of choosing the cloud,
the circle or the square is 1/3, since all three objects are blue.
The same goes for the utterance solid.
```

`determineValidUtterances` *Determine valid utterances*

Description

The relevant utterances are determined based on the features of the current objects present in the scene.

Usage

```
determineValidUtterances(currentObjects)
```

Arguments

currentObjects	A vector of three values in range {1, . . . , 27} specifying the target and the other two objects. The target is the first object in the vector (index = 1).

Value

A vector of utterances that correspond to all feature values present in the current objects in the scene.

Examples

```
currentObjects <- c(1,2,3)

determineValidUtterances(currentObjects)

output: [1] 1 2 3 4 7

To see which objects are present in the scene run:
allObjects[currentObjects,]

output:

      shape    pattern color
[1,] "cloud"  "solid" "blue"
[2,] "circle" "solid" "blue"
[3,] "square" "solid" "blue"

Based on this output you can see that the indices in the validUtterances vector refer to:
cloud = 1, circle = 2, square = 3, solid = 4 and blue = 7.
```

---

getBestInfGainUttPreferences

Get the utterances for the speaker's best information gain

Description

Full-RSA

Determines the optimal utterance for the best information gain.

These are based on the valid utterances determined from the current objects in the scene.

The inferred listener's object preferences are computed assuming the listener picks a certain object and has certain object preferences.

Usage

```
getBestInfGainUttPreferences(
  currentObjects,
  softPrefValue,
  notObeyInst,
  alpha,
  klValueFactor
)
```

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)`.
`softPrefValue`	A parameter value between `[0,infinity)` (The larger the value 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"*.
`alpha`	A parameter value between 0 and 1. Exponential scaling of the speaker choosing the utterance that maximizes the chance of the listener getting the target object right.
`klValueFactor`	A parameter that can be negative, 0 or positive: **zero** Don't care about learning about the feature preferences of the listener **positive** Care about learning about the feature preferences of the listener **negative** Trying to pick non-ambiguous utterances

Details

This function uses the function `{bestInfGainUtterance}`.

Value

A vector containing the normalized probability over utterances given the listener's object preference priors.

The utterance with the highest probability is the one that maximizes the information gain for the speaker.

The vector has the same length as the validUtterances vector.

Examples

```
In the case of these objects being in a scene:

     [shape] [pattern] [color]
[1,] "cloud"  "solid" "blue"
[2,] "circle" "solid" "blue"
[3,] "square" "solid" "blue"

and these being the indices for the valid utterances:

[1] 1 2 3 4 7 (cloud, circle, square, solid, blue)

Then uttering solid or blue would be best in order to gain information
about the shape preferences of the listener:
getBestInfGainUttPreferences(currentObjects, softPrefValue,
notObeyInst, alpha, klValueFactor)

output:
[1] 0.0 0.0 0.0 0.5 0.5

You can see here that the indices with the highest probability, namely 4 and 5,
correspond to the indices in the validUtterance vector for the feature
values 4 (solid) (index = 4) and 7 (blue) (index = 5).
```

`getConstellationCode` *Get object constellation code*

Description

This function determines the ambiguity classes. These are determined by answering this question: What properties does target object share with other objects?

This allows you to group together the trials that have the same levels of ambiguity.

Usage

```
getConstellationCode(objectConstellation, chosenFeature)
```

Arguments

`objectConstellation`	A vector of three values in range {1, . . . , 27} specifying the target and the other two objects in the scene. The target is the first object in the vector (`index = 1`).
`chosenFeature`	A value {1, 2 or 3} that specifies, which one of the three feature values (shape, texture or color) of the chosen object was used as the utterance. **Example:** utterance was *"red"*, means chosen feature is *color* (`chosenFeature = 3`).

Value

A list of three rows (length of the objectConstellation vector). Each row..

Examples

```
getConstellationCode(objectConstellation, chosenFeature)

output:
[[1]]
[1] 1 1 3 2 3 2

[[2]]
[1] 1 2 3

[[3]]
[1] 1 2 3

To see which object are in the object constellation run:

allObjects[objecConstellation,]

output:
     shape   pattern  color
[1,] "cloud"  "solid" "blue"
[2,] "circle" "solid" "blue"
[3,] "square" "solid" "blue"
```

`getObjectPreferencePriors` *Get priors of object preferences of the listener*

Description

Determine the priors of object preferences of the listener.

These are automatically derived from the valid utterances in the scene (i.e. derived from all features of the current objects).

Usage

```
getObjectPreferencePriors(validUtterances, currentObjects, type, uttToObjProbs)
```

Arguments

`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`	A vector of three values in range $\{1, \ldots, 27\}$ specifying the target and the other two objectsin the scene. The target is the first object in the vector (`index = 1`).
`type`	Is set to `0`: Hard priors or `>0`: Soft priors with specified softness

`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.

Details

This function is only used in X9

Value

A list of preference priors for all valid utterances based on the objects 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 preferences whatsoever, i.e. uniform prior over all three objects in the scene.

Examples

```
getObjectPreferencePriors(validUtterances, currentObjects, type, uttToObjProbs)

output:
[[1]]
[1] 0.85 0.077 0.077

[[2]]
[1] 0.077 0.85 0.077

[[3]]
[1] 0.077 0.077 0.85

[[4]]
[1] 0.33 0.33 0.33

[[5]]
[1] 0.33 0.33 0.33

[[6]]
[1] 0.33 0.33 0.33


Example:

To see which objects are present in the scene run:
allObjects[currentObjects,]

output:

      shape   pattern  color
[1,] "cloud"  "solid" "blue"
[2,] "circle" "solid" "blue"
[3,] "square" "solid" "blue"
```

```
The validUtterances vector: [1] 1 2 3 4 7

The rows of the list 1-6 refer to the valid utterances in the scene.

Example:

If we take a look at row 4 of the list:
[[4]]
[1] 0.33 0.33 0.33
We can see that the preference prior of the listener is 1/3 for
each of the objects, if the utterance is "solid".
The last row (6) in the list contains a uniform prior over the present
objects in the scene.
```

`getPreferencesPrior` *Get prior preferences of the listener*

Description

Simple RSA

Usage

```
getPreferencesPrior(targetFeature)
```

Arguments

`targetFeature`	A value between 1 and 3, specifying which feature type- color, shape, or pattern- is considered (for preferences).

Value

A vector of length 9. It contains a uniform prior over the three features of the specified feature type and zeros for the other feature values.

Examples

```
getPreferencesPrior(targetFeature)

output:
[1] 0.33 0.33 0.33 0.00 0.00 0.00 0.00 0.00 0.00

 @details
 This function is used in X9.
```

`getSimpleBestInfGainUttPreferences` *Get the utterances for the speaker's best information gain*

Description

Simple RSA

Determines the optimal utterance for the best information gain.

These are based on the valid utterances determined from the current objects in the scene.

The inferred listener's object preferences are computed assuming the listener picks a certain object and has certain object preferences.

Usage

```
getSimpleBestInfGainUttPreferences(
  currentObjects,
  softPrefValue,
  notObeyInst,
  klValueFactor
)
```

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)`.
`softPrefValue`	A parameter value between `[0,infinity)` (The larger the value 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"*.
`klValueFactor`	A parameter that can be negative, zero or positive: **zero** Don't care about learning about feature preferences of the listener **positive** Care about learning about feature preferences of the listener **negative** Trying to pick non-ambiguous utterances

Details

This function uses the function `simpleBestInfGainUtterance`.

Value

A vector containing the normalized probability over utterances given the listener's object preference priors.

The utterance with the highest probability is the one that maximizes the information gain for the speaker.

The vector has the same length as the validUtterances vector.

Examples

```
In the case of these objects being in a scene:

     [shape] [pattern] [color]
[1,] "cloud"  "solid" "blue"
[2,] "circle" "solid" "blue"
[3,] "square" "solid" "blue"

and these being the indices for the valid utterances:

[1] 1 2 3 4 7 (cloud, circle, square, solid, blue)

Then uttering solid or blue would be best in order to gain
information about the shape preferences of the listener:
getSimpleBestInfGainUttPreferences(currentObjects, softPrefValue,
notObeyInst, klValueFactor)

output:
[1] 0.0 0.0 0.0 0.5 0.5

You can see here that the indices with the highest probability, namely 4 and 5,
correspond to the indices in the validUtterance vector for the feature
values 4 (solid) (index = 4) and 7 (blue) (index = 5).
```

---

getSimpleBestInfGainUttPreferencesFTF

Feature Type Focus: Get the utterances for the speaker's best information gain

---

Description

Simple RSA

Determines the optimal utterance for the best information gain.

These are based on the valid utterances determined from the current objects in the scene.

The inferred listener's object preferences are computed assuming the listener picks a certain object and has certain object preferences.

Utterances of the feature type focus are filtered out (e.g. if you are interested in color preferences - so you are not allowed to utter colors!) validUtterancesFTF correspond to all features present in the current objects except those of the featureTypeFocus!

Usage

```
getSimpleBestInfGainUttPreferencesFTF(
  currentObjects,
  featureTypeFocus,
  softPrefValue,
  notObeyInst,
  klValueFactor
)
```

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)`.
`featureTypeFocus`	A value between 1-3. It specifies the feature the speaker is interested to know more about and is not allowed to utter. (shape = 1, pattern = 2, color = 3).
`softPrefValue`	A parameter value between `[0,infinity)` (The larger the value 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"*.
`klValueFactor`	A parameter that can be negative, 0 or positive: **zero** Don't care about learning about feature preferences of the listener **positive** Care about learning about feature preferences of the listener **negative** Trying to pick non-ambiguous utterances

Value

A vector containing the normalized probability over utterances given the listener's object preference priors.

The utterance with the highest probability is the one that maximizes the information gain for the speaker.

The vector has the same length as the validUtterancesFTF vector.

Examples

```
In the case of these objects being in a scene:

     [shape] [pattern] [color]
[1,] "cloud"  "solid" "blue"
```

```
[2,] "circle" "solid" "blue"
[3,] "square" "solid" "blue"

and these being the indices for the valid utterances:

[1] 1 2 3 4 7 (cloud, circle, square, solid, blue)

Since you are interested in shape (featureTypeFocus = 1) you are only
allowed to utter "solid" (index = 4) or "blue" (index = 7):

validUtterancesFTF:

[1] 4 7

Then uttering solid or blue would be best in order to gain
information about the shape preferences of the listener:
getSimpleBestInfGainUttPreferencesFTF (currentObjects, featureTypeFocus,
softPrefValue, notObeyInst, klValueFactor)

output:
[1] 0.0 0.0 0.0 0.5 0.5

You can see here that the indices with the highest probability, namely 4 and 5,
correspond to the indices in the validUtterance vector for the feature
values 4 (solid) (index = 4) and 7 (blue) (index = 5).
```

`getSimpleBestInfGainUttPreferencesIterative` *Get the utterances for the speaker's best information gain (iterative setting)*

Description

Simple RSA

Determines the optimal utterance for the best information gain.

These are based on the valid utterances determined from the current objects in the scene.

The inferred listener's object preferences are computed assuming the listener picks a certain object and has certain object preferences.

Usage

```
getSimpleBestInfGainUttPreferencesIterative(
  preferencesPriorAll,
  currentObjects,
  softPrefValue,
  notObeyInst,
  klValueFactor,
  targetFeature
)
```

Arguments

`preferencesPriorAll`
: A vector of length 9.
Probability mass over all feature values.
Gives a prior preferences distribution over all (nine) feature values.

`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)`.

`softPrefValue`
: A parameter value between `[0,infinity)` (The larger the value 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"*.

`klValueFactor`
: A parameter that can be negative, zero or positive:
zero Don't care about learning about feature preferences of the listener
positive Care about learning about feature preferences of the listener
negative Trying to pick non-ambiguous utterances

`targetFeature`
: A value between 1 and 3, specifying which feature type- color, shape, or pattern- is considered (for preferences).

Details

This function is only used in X9.

This is the iterative version of `getSimpleBestInfGainUttPreferences`

Value

A vector containing the normalized probability over utterances given the listener's object preference priors.

The utterance with the highest probability is the one that maximizes the information gain for the speaker.

posterior preferences over feature values: 3 dimensional array for simulated preferences.

rows: utterances, **columns:** preferences, **blocks**: objects.

Examples

```
getSimpleBestInfGainUttPreferencesIterative(preferencesPriorAll, currentObjects, softPrefValue,
notObeyInst, klValueFactor, targetFeature)

output:
```

```
[[1]]
[1] 0.25 0.25 0.25 0.00 0.00 0.25
[[2]]
, , 1

      [,1] [,2]  [,3]    [,4]       [,5]      [,6]       [,7] [,8] [,9]
[1,]    0    0    0    0.3334166   0.33325   0.3333333    0    0    0
[2,]    0    0    0    0.3334166   0.33325   0.3333333    0    0    0
[3,]    0    0    0    0.3334167   0.33325   0.3333333    0    0    0
[4,]    0    0    0    0.0000000   0.00000   0.0000000    0    0    0
[5,]    0    0    0    0.0000000   0.00000   0.0000000    0    0    0
[6,]    0    0    0    0.3334166   0.33325   0.3333333    0    0    0

, , 2

      [,1] [,2] [,3]    [,4]       [,5]      [,6]     [,7] [,8] [,9]
[1,]    0    0    0    0.3334166   0.33325   0.3333333    0    0    0
[2,]    0    0    0    0.3334166   0.33325   0.3333333    0    0    0
[3,]    0    0    0    0.3334167   0.33325   0.3333333    0    0    0
[4,]    0    0    0    0.0000000   0.00000   0.0000000    0    0    0
[5,]    0    0    0    0.0000000   0.00000   0.0000000    0    0    0
[6,]    0    0    0    0.3334166   0.33325   0.3333333    0    0    0

, , 3

      [,1] [,2] [,3]     [,4]      [,5]      [,6]      [,7] [,8] [,9]
[1,]    0    0    0    0.3331667  0.3334999  0.3333333    0    0    0
[2,]    0    0    0    0.3331667  0.3334999  0.3333333    0    0    0
[3,]    0    0    0    0.3331668  0.3334999  0.3333333    0    0    0
[4,]    0    0    0    0.0000000  0.0000000  0.0000000    0    0    0
[5,]    0    0    0    0.0000000  0.0000000  0.0000000    0    0    0
[6,]    0    0    0    0.3331667  0.3334999  0.3333333    0    0    0
```

`getSpeakerUtterancePriors` *Get the speaker's utterances' priors*

Description

Full-RSA

This function determines the prior utterance preferences of the speaker.

Usage

```
getSpeakerUtterancePriors(validUtterances)
```

Arguments

`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.

Value

A vector of priors as numeric values.

Examples

```
getSpeakerUtterancePriors(validUtterances)

output:
[1] 0.2 0.2 0.2 0.2 0.2
```

`getSpeakerUtteranceUniformPrior`
Speaker's uniform priors for utterances

Description

Simple RSA

Determines the prior utterance preferences of the speaker.

Usage

```
getSpeakerUtteranceUniformPrior(validUtterances)
```

Arguments

`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.

Value

A vector of the same length as the validUtterances vector.

It contains numeric values of the prior utterance preferences of the speaker.

Examples

```
getSpeakerUtteranceUniformPrior(validUtterances)

output:
 [1] 0.17 0.17 0.17 0.17 0.17 0.17

 @details
 This function is used in X9.
```

`getUtteranceChoiceConstellationCode` *Get utterance-choice's constellation code*

Description

Method that determines the utterance choice constellation code.

It also determines the resulting feature-order, the object-order and the feature-value-order.

The feature-order depends on the individual feature value ambiguities.

Usage

```
getUtteranceChoiceConstellationCode(objectConstellation)
```

Arguments

`objectConstellation`
: A vector of three values in range {1, . . . , 27} specifying the target and the other two objects in the scene.
The target is the first object in the vector (index = 1).

Value

A list of four rows.

First row: resultCode

Second row: featureOrder

Thrid row: objectOrder

Fourth row: featureValueOrder

Examples

```
getUtteranceChoiceConstellationCode(objectConstellation)

output:
[[1]]
[1] "331"

[[2]]
[1] 3 2 1

[[3]]
[1] 1 2 3

[[4]]
[1] 7 4 1 2 3

To see which object are in the object constellation run:
allObjects[objectConstellation,]

output:
     shape   pattern  color
```

```
[1,] "cloud"  "solid" "blue"
[2,] "circle" "solid" "blue"
[3,] "square" "solid" "blue"
```

`KLdivergence` *Calculate Kullback-Leibler divergence*

Description

Divergence measure.

Simple KL divergence function- with small offset to tolerate `p = 0 / q = 0`

Usage

```
KLdivergence(p, q)
```

Arguments

`p`	Probability distribution.
`q`	Probability distribution.

Details

Full-RSA and Simple-RSA

Value

A scalar value (The divergence between p and q).

`literalListener` *Literal listener function*

Description

Full-RSA

Literal listener function according to assigned listener's object preferences.

Usage

```
literalListener(utterance, listenerObjectPreferences, uttToObjProbs)
```

Arguments

`utterance`	The uttered word by the speaker that the listener hears. An index referring to one of the values in the vector validUtterances.
`listenerObjectPreferences`	One of the rows of the 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.
`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.

Value

P(obj | utt, listener's object preferences)

A Vector of length 3. It includes the normalized probability of choosing each of the three objects in the scene, given the utterance and the listener's object preferences.

Examples

```
listenerObjectPreferences <- objectPreferenceSoftPriors[[3]]
literalListener(utterance, listenerObjectPreferences, uttToObjProbs)

output:
[1] 1 0 0
```

`mapObjectToUtterances` *Map objects to utterances*

Description

Determining the specific mapping of objects to utterances that applies given the present objects in the scene.

Usage

```
mapObjectToUtterances(currentObjects)
```

Arguments

`currentObjects`	A vector of three values in range `{1,...,27}` specifying the target and the other two objects. The target is the first object in the vector (`index = 1`).

Details

This function is only used in X9

Value

A 3x3 matrix. Values in the matrix are in range {1-9}.

rows: Objects; **columns:** Feature values of the objects

Examples

```
currentObjects <- c(1,2,3)
mapObjectToUtterances(currentObjects)

output:
      [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    4    7
[3,]    3    4    7

To see which objects are present in the scene run:
allObjects[currentObjects,]

output:

      shape   pattern  color
[1,] "cloud"  "solid" "blue"
[2,] "circle" "solid" "blue"
[3,] "square" "solid" "blue"

Based on this output you can see that the indices output matrix vector refer
to the following feature values:
cloud = 1, circle = 2, square = 3, solid = 4 and blue = 7.
```

---

pragmaticListener *Pragmatic listener function samples over the speaker*

Description

Full-RSA

Determines the probability of the listener choosing a certain object given the speaker's utterance and her own object preferences.

P(obj | utt, listener's object preferences).

It essentially infers which object it should pick under a certain literal listener preference.

Usage

```
pragmaticListener(
  utterance,
  listenerObjectPreferences,
  validUtterances,
```

```
  currentObjects,
  uttToObjProbs,
  alpha
)
```

Arguments

`utterance`	The uttered word by the speaker that the listener hears. An index referring to one of the values in the vector validUtterances.
`listenerObjectPreferences`	One of the rows of the 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.
`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.
`alpha`	A parameter between 0 and 1. Exponential scaling of the speaker choosing the utterance that maximizes the chance of the listener getting the target object right.

Value

A vector of the normalized probability of the listener choosing a certain object given the speaker's utterance and her own object preferences.

Examples

```
listenerObjectPreferences <- objectPreferenceSoftPriors[[2]]
pragmaticListener(utterance, listenerObjectPreferences,
validUtterances, currentObjects, uttToObjProbs, alpha)

output:
[1] 1 0 0

To see which object are in the object constellation run:
```

```
allObjects[currentObjects,]

output:
      [,1]     [,2]    [,3]
[1,] "cloud"  "solid" "blue"
[2,] "circle" "solid" "blue"
[3,] "square" "solid" "blue"
```

`pragmaticSpeaker` *Pragmatic Speaker function*

Description

Full-RSA

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

It starts with a prior assumption over the possible preferences of the listener. Then it infers the posterior over these preferences given an object choice of the listener i.e. P(listener's feature value preferences | utterance, object choice by the listener, prior over preferences)

Usage

```
pragmaticSpeaker(
  utterance,
  obj,
  preferencesPrior,
  validUtterances,
  currentObjects,
  uttToObjProbs,
  objectPreferenceSoftPriors,
  alpha
)
```

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.
`preferencesPrior`	A vector of length 9. Probability mass over all feature values present in the scenario plus a "no preference" case. Gives a prior preferences distribution over all (nine) feature values.
`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` 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).

`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.

`alpha` A parameter between 0 and 1.

Exponential scaling of the speaker choosing the utterance that maximizes the chance of the listener getting the target object right.

Value

A vector with the same as the validUtterances vector.

Normalized posterior probability over preferences- given the utterance, the object choice by the listener, and prior over preferences of the listener.

Examples

```
pragmaticSpeaker(utterance, obj, preferencesPrior, validUtterances,
currentObjects, uttToObjProbs, objectPreferenceSoftPriors, alpha)

output:
[1] 0.17 0.17 0.17 0.17 0.17 0.17
```

RSAModelKLDiv3paramsAllValuesConsidered

RSA model Kullback-Leibler divergence determination (all feature values considered)

Description

Full-RSA

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.

3 parameter optimization considering all feature values (also the ones not present in the scene), i.e. feature values of shape, texture and color.

Usage

```
RSAModelKLDiv3paramsAllValuesConsidered(data, par1, par2, par3)
```

Arguments

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.
par1	**softness parameter** Specifies how much actual feature priorities come into play in the object choice (The larger 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	**alpha parameter** An exponential scaling of the speaker choosing the utterance that maximizes the chance of the listener getting the target object right.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`RSAModelKLDiv3paramsAllValuesConsidered_simpleRSA`
Simple RSA model Kullback-Leibler divergence determination (all feature values considered)

Description

Simple RSA

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.

3 parameter optimization considering all feature values (also the ones not present in the scene), i.e. feature values of shape, texture and color.

Usage

```
RSAModelKLDiv3paramsAllValuesConsidered_simpleRSA(data, par1, par2)
```

Arguments

`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.
`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).

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`RSAModelKLDiv3paramsOnlyAvailableFeatureValuesConsidered` *RSA model Kullback-Leibler divergence determination (available feature values only)*

Description

Full-RSA

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.

3 parameter optimization considering only the available feature values present in the scene, i.e. feature values of shape, texture and color.

Usage

```
RSAModelKLDiv3paramsOnlyAvailableFeatureValuesConsidered(
  data,
  par1,
  par2,
  par3
)
```

Arguments

`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.
`par1`	**softness parameter** Specifies how much actual feature priorities come into play in the object choice (The larger 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 **alpha parameter** An exponential scaling of the speaker choosing the utterance that maximizes the chance of the listener getting the target object right.

Details

This function is used in the functions RSAModelLL1_1, RSAModelLL1_1_notObey.1, RSAModelLL1_2, RSAModelLL1_3, RSAModelLL2_n1, RSAModelLL2_n2, RSAModelLL2_n3, RSAModelLL3.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

RSAModelKLDiv3paramsOnlyAvailableFeatureValuesConsidered_simpleRSA

Simple RSA model Kullback-Leibler divergence determination (available feature values only)

Description

Simple RSA

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.

3 parameter optimization considering only the available feature values present in the scene, i.e. feature values of shape, texture and color.

Usage

```
RSAModelKLDiv3paramsOnlyAvailableFeatureValuesConsidered_simpleRSA(
  data,
  par1,
  par2
)
```

Arguments

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.

`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 `RSAModelLL1_1simpleRSA`, `RSAModelLL1_1simpleRSA_notObey.1`, `RSAModelLL1_1simpleRSA_notObey.2`, `RSAModelLL1_2simpleRSA`, `RSAModelLL1_2simpleRSA_pref2`, `RSAModelLL2_simpleRSA`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`RSAModelKLDiv3params_simpleRSA4TrialsIndependent`

Simple RSA model Kullback-Leibler divergence determination (for independent trials)

Description

Simple RSA

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.

Usage

```
RSAModelKLDiv3params_simpleRSA4TrialsIndependent(data, par1, par2)
```

Arguments

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.
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 RSAModelLL1_1simpleRSA4TrialsIndependent,

RSAModelLL1_1simpleRSA4TrialsIndependent_notObey.1,

RSAModelLL1_1simpleRSA4TrialsIndependent_notObey.2,

RSAModelLL1_2simpleRSA4TrialsIndependent,

RSAModelLL1_2simpleRSA4TrialsIndependent_pref.2,

RSAModelLL2_simpleRSA4TrialsIndependent.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`RSAModelKLDiv3params_simpleRSA4TrialsIterative` *Simple RSA model Kullback-Leibler divergence determination (iterative setting)*

Description

Simple RSA

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.

Usage

```
RSAModelKLDiv3params_simpleRSA4TrialsIterative(data, par1, par2)
```

Arguments

`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.
`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 RSAModelLL1_1simpleRSA4TrialsIterative,
RSAModelLL1_1simpleRSA4TrialsIterative_notObey.1,
RSAModelLL1_1simpleRSA4TrialsIterative_notObey.2,
RSAModelLL1_2simpleRSA4TrialsIterative,
RSAModelLL1_2simpleRSA4TrialsIterative_pref.2,
RSAModelLL2_simpleRSA4TrialsIterative.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`RSAModelLL1_1` *Cost function for one parameter optimization. Optimizing softness. Non-obedience fixed at 0.*

Description

Full-RSA

1 parameter optimization; The softness parameter is optimized.

The non-obedience parameter and the alpha parameter are fixed.

Usage

```
RSAModelLL1_1(params, data)
```

Arguments

`params`	One value vector, which specifies one of three parameters to be optimized: 1. softPrefValue parameter 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 parameter 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. alpha parameter is fixed at 1, i.e. An exponential scaling of the speaker choosing the utterance that maximizes the chance of the listener getting the target object right.
`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 `RSAModelKLDiv3paramsOnlyAvailableFeatureValuesConsidered`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`RSAModelLL1_1simpleRSA`

Cost function for one parameter optimization. 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

```
RSAModelLL1_1simpleRSA(params, data)
```

Arguments

`params`	One value vector specifying one of the two parameters to be optimized: 1. softPrefValue parameter 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 parameter 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).
`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 `RSAModelKLDiv3paramsOnlyAvailableFeatureValuesConsidered_simpleRSA`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`RSAModelLL1_1simpleRSA4TrialsIndependent`

Cost function for one parameter optimization (for independent trials). Optimizing softness. Non-obedience is fixed at 0.

Description

Simple RSA

1 parameter optimization; The softness parameter is optimized.

The non-obedience parameter is fixed.

Usage

```
RSAModelLL1_1simpleRSA4TrialsIndependent(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 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)

`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_simpleRSA4TrialsIndependent`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`RSAModelLL1_1simpleRSA4TrialsIndependent_notObey.1`

Cost function for one parameter optimization (for independent trials). Optimizing softness. Non-obedience is fixed at 0.1.

Description

Simple RSA

1 parameter optimization; The softness parameter is optimized.

The non-obedience parameter is fixed.

Usage

```
RSAModelLL1_1simpleRSA4TrialsIndependent_notObey.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 is 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_simpleRSA4TrialsIndependent`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`RSAModelLL1_1simpleRSA4TrialsIndependent_notObey.2`

Cost function for one parameter optimization (for independent trials). Optimizing softness. Non-obedience is fixed at 0.2.

Description

Simple RSA

1 parameter optimization; The softness parameter is optimized.

The non-obedience parameter is fixed.

Usage

```
RSAModelLL1_1simpleRSA4TrialsIndependent_notObey.2(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 fixed at 0.2, 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_simpleRSA4TrialsIndependent`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`RSAModelLL1_1simpleRSA4TrialsIterative` *Cost function for one parameter optimization (iterative setting). 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

```
RSAModelLL1_1simpleRSA4TrialsIterative(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`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`RSAModelLL1_1simpleRSA4TrialsIterative_notObey.1`
Cost function for one parameter optimization (iterative setting). 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

```
RSAModelLL1_1simpleRSA4TrialsIterative_notObey.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 is 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.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`RSAModelLL1_1simpleRSA4TrialsIterative_notObey.2`

Cost function for one parameter optimization (iterative setting). Optimizing softness. Non-obedience fixed at at 0.2

Description

Simple RSA

1 parameter optimization; The softness parameter is optimized.

The non-obedience parameter is fixed.

Usage

```
RSAModelLL1_1simpleRSA4TrialsIterative_notObey.2(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 fixed at 0.2, 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.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`RSAModelLL1_1simpleRSA_notObey.1`

Cost function for one parameter optimization. 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

```
RSAModelLL1_1simpleRSA_notObey.1(params, data)
```

Arguments

`params`	One value vector specifying one of the two parameters to be optimized: 1. softPrefValue parameter 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 0.1, 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 `RSAModelKLDiv3paramsOnlyAvailableFeatureValuesConsidered_simpleRSA`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`RSAModelLL1_1simpleRSA_notObey.2`

Cost function for one parameter optimization. Optimizing softness. Non-obedience fixed at 0.2.

Description

Simple RSA

1 parameter optimization; The softness parameter is optimized.

The non-obedience parameter is fixed.

Usage

```
RSAModelLL1_1simpleRSA_notObey.2(params, data)
```

Arguments

`params`	One value vector specifying one of the two parameters to be optimized: 1. softPrefValue parameter 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 0.2, 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 `RSAModelKLDiv3paramsOnlyAvailableFeatureValuesConsidered_simpleRSA`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`RSAModelLL1_1_notObey.1`

Cost function for one parameter optimization. Optimizing softness. Non-obedience fixed at 0.1.

Description

Full-RSA

1 parameter optimization; The softness parameter is optimized.

The non-obedience parameter and the alpha parameter are fixed.

Usage

```
RSAModelLL1_1_notObey.1(params, data)
```

Arguments

`params`	One value vector specifying one of the three parameters to be optimized: 1. softPrefValue parameter 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 parameter is 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)

3. alpha parameter is fixed at 1, i.e. An exponential scaling of the speaker choosing the utterance that maximizes the chance of the listener getting the target object right.

`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 `RSAModelKLDiv3paramsOnlyAvailableFeatureValuesConsidered`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`RSAModelLL1_2` *Cost function for one parameter optimization. Optimizing non-obedience. Softness fixed at 0.*

Description

Full-RSA

1 parameter optimization; The non-obedience parameter is optimized.

The softness parameter and the alpha parameter are set to certain values.

Usage

```
RSAModelLL1_2(params, data)
```

Arguments

`params` One value vector specifying one of the three parameters to be optimized:

1. softPrefValue parameter 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 parameter 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. alpha parameter is fixed at 1, i.e. An exponential scaling of the speaker choosing the utterance that maximizes the chance of the listener getting the target object right.

`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 `RSAModelKLDiv3paramsOnlyAvailableFeatureValuesConsidered`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`RSAModelLL1_2simpleRSA`

Cost function for one parameter optimization. Optimizing non-obedience. Softness is fixed at 0.2.

Description

Simple RSA

1 parameter optimization; The non-obedience parameter is optimized.

The softness parameter is fixed.

Usage

```
RSAModelLL1_2simpleRSA(params, data)
```

Arguments

`params`	One value vector specifying one of the two parameters to be optimized: 1. softPrefValue parameter 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 parameter 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 `RSAModelKLDiv3paramsOnlyAvailableFeatureValuesConsidered_simpleRSA`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`RSAModelLL1_2simpleRSA4TrialsIndependent` *Cost function for one parameter optimization (for independent trials). Optimizing non-obedience. Softness is fixed at 0.*

Description

Simple RSA

1 parameter optimization; The non-obedience parameter is optimized.

The softness parameter is fixed.

Usage

```
RSAModelLL1_2simpleRSA4TrialsIndependent(params, data)
```

Arguments

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

1. softPrefValue is 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)

.

`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_simpleRSA4TrialsIndependent.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

RSAModelLL1_2simpleRSA4TrialsIndependent_pref.2
Cost function for one parameter optimization (for independent trials). Optimizing non-obedience. Softness is fixed at 0.2.

Description

Simple RSA

1 parameter optimization; The non-obedience parameter is optimized.

The softness parameter is fixed.

Usage

```
RSAModelLL1_2simpleRSA4TrialsIndependent_pref.2(params, data)
```

Arguments

params	One value vector, which specifies one of two parameters to be optimized: 1. softPrefValue is is fixed at 0.2, 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, 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_simpleRSA4TrialsIndependent`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`RSAModelLL1_2simpleRSA4TrialsIterative`

Cost function for one parameter optimization (iterative setting). Optimizing non-obedience. Softness fixed at at 0.

Description

Simple RSA

1 parameter optimization; non-obedience parameter is optimized.

The softness parameter is fixed.

Usage

```
RSAModelLL1_2simpleRSA4TrialsIterative(params, data)
```

Arguments

`params` One value vector, which specifies one of two 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)

`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`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`RSAModelLL1_2simpleRSA4TrialsIterative_pref.2`

Cost function for one parameter optimization (iterative setting). Optimizing non-obedience. Softness fixed at at 0.2.

Description

Simple RSA

1 parameter optimization; non-obedience parameter is optimized.

The softness parameter is fixed.

Usage

```
RSAModelLL1_2simpleRSA4TrialsIterative_pref.2(params, data)
```

Arguments

`params`	One value vector, which specifies one of two parameters to be optimized: 1. softPrefValue is fixed at 0.2, 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`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`RSAModelLL1_2simpleRSA_pref2` *Cost function for one parameter optimization. Optimizing non-obedience. Softness is fixed at 0.2.*

Description

Simple RSA

1 parameter optimization; The non-obedience parameter is optimized.

The softness parameter is fixed.

Usage

```
RSAModelLL1_2simpleRSA_pref2(params, data)
```

Arguments

`params`	One value vector specifying one of the two parameters to be optimized: 1. softPrefValue is fixed at 0.2, i.e. The strength of "preferring one entity over others". (The larger the value the higher the tendency towards uniform liking) 2. non-obedience parameter 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 `RSAModelKLDiv3paramsOnlyAvailableFeatureValuesConsidered_simpleRSA`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`RSAModelLL1_3` *Cost function for one parameter optimization. Optimizing alpha. Softness and non-obedience fixed at 0.*

Description

Full-RSA

1 parameter optimization; The alpha parameter is optimized.

The softness parameter and the non-obedience parameter are fixed.

Usage

```
RSAModelLL1_3(params, data)
```

Arguments

params	One value vector specifying one of the three parameters to be optimized: 1. softPrefValue parameter 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 parameter 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. alpha parameter is optimized, i.e. An exponential scaling of the speaker choosing the utterance that maximizes the chance of the listener getting the target object right.
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 RSAModelKLDiv3paramsOnlyAvailableFeatureValuesConsidered.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

RSAModelLL2_n1 *Cost function for two parameter optimization. Optimizing non-obedience and alpha. Softness fixed at 0.*

Description

Full-RSA

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

The softness parameter is fixed.

Usage

```
RSAModelLL2_n1(params, data)
```

Arguments

`params`	Two value vector, which specifies two of three parameters (n1=not the first, n2=not the second, n3= not the third) to be optimized: 1. softPrefValueparameter 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 parameter 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. alpha parameter is optimized, i.e. An exponential scaling of the speaker choosing the utterance that maximizes the chance of the listener getting the target object right.
`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 `RSAModelKLDiv3paramsOnlyAvailableFeatureValuesConsidered`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`RSAModelLL2_n2` *Cost function for two parameter optimization. Optimizing softness and alpha. Non-obedience fixed at 0.*

Description

Full-RSA

2 parameter optimization; The softness and alpha parameters are optimized.

The non-obedience parameter is fixed.

Usage

```
RSAModelLL2_n2(params, data)
```

Arguments

`params` Two value vector, which specifies two of three parameters (n1 = not the first, n2 = not the second, n3 = not the third) to be optimized:

1. softPrefValue parameter 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 parameter 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. alpha parameter is optimized, i.e. An exponential scaling of the speaker choosing the utterance that maximizes the chance of the listener getting the target object right.

`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 `RSAModelKLDiv3paramsOnlyAvailableFeatureValuesConsidered`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`RSAModelLL2_n3` *Cost function for two parameter optimization. Optimizing softness and non-obedience. Alpha fixed at 1.*

Description

Full-RSA

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

The alpha parameter is fixed.

Usage

```
RSAModelLL2_n3(params, data)
```

Arguments

`params`	Two value vector, which specifies two of three parameters (n1 = not the first, n2 = not the second, n3 = not the third) to be optimized: 1. softPrefValue parameter 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 parameter 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. alpha parameter is fixed at 1, i.e. An exponential scaling of the speaker choosing the utterance that maximizes the chance of the listener getting the target object right.
`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 `RSAModelKLDiv3paramsOnlyAvailableFeatureValuesConsidered`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`RSAModelLL2_simpleRSA` *Cost function for two parameter optimization. Optimizing softness and non-obedience.*

Description

Simple RSA

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

Usage

```
RSAModelLL2_simpleRSA(params, data)
```

Arguments

`params` Two value vector specifying two of the two parameters to be optimized:

1. softPrefValue parameter 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 parameter is 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 `RSAModelKLDiv3paramsOnlyAvailableFeatureValuesConsidered_simpleRSA`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`RSAModelLL2_simpleRSA4TrialsIndependent`

Cost function for two parameter optimization (for independent trials). Optimizing softness and non-obedience.

Description

Simple RSA

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

Usage

```
RSAModelLL2_simpleRSA4TrialsIndependent(params, data)
```

Arguments

`params` Two value vector, which specifies two 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_simpleRSA4TrialsIndependent`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`RSAModelLL2_simpleRSA4TrialsIterative`

Cost function for one parameter optimization (iterative setting). Optimizing softness and non-obedience.

Description

Simple RSA

2 parameter optimization; softPrefValue and non-obedience are optimized.

Usage

```
RSAModelLL2_simpleRSA4TrialsIterative(params, data)
```

Arguments

`params` Two value vector, which specifies two 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`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`RSAModelLL3` *Cost function for three parameter optimization. Optimizing softness, non-obedience and alpha.*

Description

Full-RSA

3 parameter optimization; The softness parameter, the non-obedience and the alpha parameter are optimized.

Usage

```
RSAModelLL3(params, data)
```

Arguments

params	Three value vector, which specifies all three model parameters to be optimized: 1. softPrefValue parameter 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 parameter 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. alpha parameter is optimized, i.e. An exponential scaling of the speaker choosing the utterance that maximizes the chance of the listener getting the target object right.
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 RSAModelKLDiv3paramsOnlyAvailableFeatureValuesConsidered.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`RSAModelUttKLDivParamA` *Cost function for one parameter optimization for the utterance choice experiments. Optimizing softness. Non-obedience and klValueFactor are fixed.*

Description

Full-RSA

Softness is optimized.

Non-obedience is fixed at 0 and klValueFactor at 1.

Usage

```
RSAModelUttKLDivParamA(par, data)
```

Arguments

`par`	One value vector, which specifies one of three parameters to be optimized: 1. softPrefValue: A parameter value between `[0,infinity)` (The larger the value 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.
`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.

Details

This function uses `RSAModelUttKLDiv_3params`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`RSAModelUttKLDivParamAB` *Cost function for two parameter optimization for the utterance choice experiments. Optimizing softness and non-obedience. klValueFactor is fixed.*

Description

Full-RSA

Softness and non-obedience are optimized.

klValueFactor is fixed at 1.

Usage

```
RSAModelUttKLDivParamAB(params, data)
```

Arguments

`params`	Two value vector specifying the two parameters to be optimized. 1. softness A parameter value between `[0,infinity)` (The larger the value 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. 2. non-obedience: 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"*.
`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.

Details

This function uses `RSAModelUttKLDiv_3params`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`RSAModelUttKLDivParamABC` *Cost function for three parameter optimization for the utterance choice experiments. Optimizing softness, non-obedience and klValueFactor.*

Description

Full-RSA

Softness, non-obedience and klValueFactor are optimized.

Usage

```
RSAModelUttKLDivParamABC(params, data)
```

Arguments

`params` Three value vector specifying the three parameters to be optimized.

1. softness: A parameter value between `[0,infinity)` (The larger the value 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.
2. non-obedience: 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"*.
3. klValueFactor: A parameter that can be negative, 0 or positive (Here it is set to = 1):

   zero Don't care about learning about the feature preferences of the listener

   positive Care about learning about the feature preferences of the listener

   negative Trying to pick non-ambiguous utterances

`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.

Details

This function uses `RSAModelUttKLDiv_3params`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`RSAModelUttKLDivParamABCD4`
Cost function for four parameter optimization for the utterance choice experiments. Optimizing softness, non-obedience, alpha and klValue-Factor.

Description

Full-RSA

Softness, non-obedience, alpha and klValueFactor are optimized.

Usage

```
RSAModelUttKLDivParamABCD4(params, data)
```

Arguments

`params` Four value vector specifying four of the four parameters to be optimized:

1. softPrefValue: A parameter value between `[0,infinity)` (The larger the value 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.
2. non-obedience: 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"*.
3. alpha: A parameter value between 0 and 1.
   Exponential scaling of the speaker choosing the utterance that maximizes the chance of the listener getting the target object right.
4. klValueFactor can be negative, zero or positive:
   zero Don't care about learning about feature preferences of the listener
   positive Care about learning about feature preferences of the listener
   negative Trying to pick non-ambiguous utterances

`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.

Details

This function uses `RSAModelUttKLDiv_4params`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`RSAModelUttKLDivParamABD4`

Cost function for three parameter optimization for the utterance choice experiments. Optimizing softness, non-obedience and klValueFactor. Alpha is fixed.

Description

Full-RSA

Softness, non-obedience and klValueFactora re optimized.

Alpha is fixed at 1.

Usage

```
RSAModelUttKLDivParamABD4(params, data)
```

Arguments

`params` Three value vector specifying three of the four parameters to be optimized:

1. softPrefValue: A parameter value between `[0,infinity)` (The larger the value 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.
2. non-obedience: 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"*.
3. klValueFactor can be negative, zero or positive:
   zero Don't care about learning about feature preferences of the listener
   positive Care about learning about feature preferences of the listener
   negative Trying to pick non-ambiguous utterances

`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.

Details

This function uses `RSAModelUttKLDiv_4params`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`RSAModelUttKLDivParamAC`

Cost function for two parameter optimization for the utterance choice experiments. Optimizing softness and klValueFactor. Non-obedience is fixed.

Description

Full-RSA

Softness and klValueFactor are optimized.

Non-obedience is fixed at 0.

Usage

```
RSAModelUttKLDivParamAC(params, data)
```

Arguments

`params` Two value vector specifying the two parameters to be optimized.

1. softness: A parameter value between `[0,infinity)` (The larger the value 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.
2. klValueFactor: A parameter that can be negative, 0 or positive (Here it is set to = 1):
   zero Don't care about learning about the feature preferences of the listener
   positive Care about learning about the feature preferences of the listener
   negative Trying to pick non-ambiguous utterances

`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.

Details

This function uses RSAModelUttKLDiv_3params.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`RSAModelUttKLDivParamACD4`

Cost function for three parameter optimization for the utterance choice experiments. Optimizing softness, alpha and klValueFactor. Non-obedience is fixed.

Description

Full-RSA

Softness, alpha and klValueFactor are optimized.

Non-obedience is fixed at 0.

Usage

```
RSAModelUttKLDivParamACD4(params, data)
```

Arguments

`params` Three value vector specifying three of the four parameters to be optimized:

1. softPrefValue: A parameter value between `[0,infinity)` (The larger the value 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.
2. alpha: A parameter value between 0 and 1.
   Exponential scaling of the speaker choosing the utterance that maximizes the chance of the listener getting the target object right.
3. klValueFactor can be negative, zero or positive:
   zero Don't care about learning about feature preferences of the listener
   positive Care about learning about feature preferences of the listener
   negative Trying to pick non-ambiguous utterances

`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.

Details

This function uses RSAModelUttKLDiv_4params.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

RSAModelUttKLDivParamAD4

Cost function for two parameter optimization for the utterance choice experiments. Optimizing Softness and klValueFactor. Non-obedience and alpha are fixed.

Description

Full-RSA

Softness and klValueFactor are optimized.

Non-obedience is set to 0.

Alpha is set to 1.

Usage

```
RSAModelUttKLDivParamAD4(params, data)
```

Arguments

params	Two value vector specifying two of the four parameters to be optimized: 1. softPrefValue: A parameter value between [0,infinity) (The larger the value 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. 2. klValueFactor can be negative, zero or positive: **zero** Don't care about learning about feature preferences of the listener **positive** Care about learning about feature preferences of the listener **negative** Trying to pick non-ambiguous utterances
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.

Details

This function uses RSAModelUttKLDiv_4params.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`RSAModelUttKLDivParamAD4B.2`

Cost function for two parameter optimization for the utterance choice experiments. Optimizing softness and klValueFactor. Non-obedience and alpha are fixed.

Description

Full-RSA

Softness and klValueFactor are optimized.

Non-obedience is fixed at 0.2.

Alpha is fixed at 1.

Usage

```
RSAModelUttKLDivParamAD4B.2(params, data)
```

Arguments

`params`	Two value vector specifying two of the four parameters to be optimized: 1. softPrefValue: A parameter value between `[0,infinity)` (The larger the value 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. 2. klValueFactor can be negative, zero or positive: **zero** Don't care about learning about feature preferences of the listener **positive** Care about learning about feature preferences of the listener **negative** Trying to pick non-ambiguous utterances
`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.

Details

This function uses `RSAModelUttKLDiv_4params`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

RSAModelUttKLDivParamB

Cost function for one parameter optimization for the utterance choice experiments. Optimizing non-obedience. Softness and klValueFactor are fixed.

Description

Full-RSA

Non-obedience is optimized.

Softness is fixed at 0 and klValueFactor at 1.

Usage

```
RSAModelUttKLDivParamB(par, data)
```

Arguments

par	One value vector, which specifies one of three parameters to be optimized: 1. non-obedience: This parameter 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"*.
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.

Details

This function uses RSAModelUttKLDiv_3params.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`RSAModelUttKLDivParamBC` *Cost function for two parameter optimization for the utterance choice experiments. Optimizing non-obedience and klValueFactor. Softness is fixed.*

Description

Full-RSA

Non-obedience and klValueFactor are optimized.

Softness is fixed at 0.

Usage

```
RSAModelUttKLDivParamBC(params, data)
```

Arguments

`params`	Two value vector specifying the two parameters to be optimized. 1. non-obedience: 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"*. 2. klValueFactor can be negative, zero or positive: **zero** Don't care about learning about feature preferences of the listener **positive** Care about learning about feature preferences of the listener **negative** Trying to pick non-ambiguous utterances
`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.

Details

This function uses `RSAModelUttKLDiv_3params`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

RSAModelUttKLDivParamBCD4

Cost function for three parameter optimization for the utterance choice experiments. Optimizing non-obedience, alpha and klValueFactor. Softness is fixed.

Description

Full-RSA

Non-obedience, alpha and klValueFactor are optimized.

Softness is fixed at 0.

Usage

```
RSAModelUttKLDivParamBCD4(params, data)
```

Arguments

params	Three value vector specifying three of the four parameters to be optimized: 1. non-obedience: 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"*. 2. alpha: A parameter value between 0 and 1. Exponential scaling of the speaker choosing the utterance that maximizes the chance of the listener getting the target object right. 3. klValueFactor can be negative, zero or positive: **zero** Don't care about learning about feature preferences of the listener **positive** Care about learning about feature preferences of the listener **negative** Trying to pick non-ambiguous utterances
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.

Details

This function uses RSAModelUttKLDiv_4params.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`RSAModelUttKLDivParamBD4` *Cost function for two parameter optimization for the utterance choice experiments. Optimizing non-obedience and klValueFactor. Softness and alpha are fixed.*

Description

Full-RSA

Non-obedience and klValueFactor are optimized.

Softness is fixed at 0.

Alpha is fixed at 1.

Usage

```
RSAModelUttKLDivParamBD4(params, data)
```

Arguments

`params`	Two value vector specifying two of the four parameters to be optimized: 1. non-obedience: 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"*. 2. klValueFactor can be negative, zero or positive: **zero** Don't care about learning about feature preferences of the listener **positive** Care about learning about feature preferences of the listener **negative** Trying to pick non-ambiguous utterances
`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.

Details

This function uses `RSAModelUttKLDiv_4params`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`RSAModelUttKLDivParamBD4A.2` *Cost function for two parameter optimization for the utterance choice experiments. Optimizing non-obedience and klValueFactor. Softness and alpha are fixed.*

Description

Full-RSA

Non-obedience and klValueFactor are optimized.

Softness is fixed at 0.2.

Alpha is fixed at 1.

Usage

```
RSAModelUttKLDivParamBD4A.2(params, data)
```

Arguments

`params` Two value vector specifying two of the four parameters to be optimized:

1. non-obedience: 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"*.
2. klValueFactor can be negative, zero or positive:
zero Don't care about learning about feature preferences of the listener
positive Care about learning about feature preferences of the listener
negative Trying to pick non-ambiguous utterances

`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.

Details

This function uses `RSAModelUttKLDiv_4params`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

RSAModelUttKLDivParamC *Cost function for one parameter optimization for the utterance choice experiments. Optimizing klValueFactor. Softness and non-obedience are fixed.*

Description

Full-RSA

klValueFactor is optimized.

softness and non-obedience parameter are fixed at 0.

Usage

```
RSAModelUttKLDivParamC(par, data)
```

Arguments

par	One value vector, which specifies one of three parameters to be optimized: 1. klValueFactor can be negative, zero or positive: **zero** Don't care about learning about feature preferences of the listener **positive** Care about learning about feature preferences of the listener **negative** Trying to pick non-ambiguous utterances
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.

Details

This function uses RSAModelUttKLDiv_3params.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

RSAModelUttKLDivParamCD4 *Cost function for two parameter optimization for the utterance choice experiments. Optimizing alpha and klValueFactor. Softness and non-obedience are fixed.*

Description

Full-RSA

Alpha and klValueFactor are optimized.

Softness and non-obedience are fixed at 0.

Usage

```
RSAModelUttKLDivParamCD4(params, data)
```

Arguments

params	Two value vector specifying two of the four parameters to be optimized: 1. alpha: A parameter value between 0 and 1. Exponential scaling of the speaker choosing the utterance that maximizes the chance of the listener getting the target object right. 2. klValueFactor: can be negative, zero or positive: **zero** Don't care about learning about feature preferences of the listener **positive** Care about learning about feature preferences of the listener **negative** Trying to pick non-ambiguous utterances
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.

Details

This function uses RSAModelUttKLDiv_4params.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

RSAModelUttKLDivParamD4 *Cost function for one parameter optimization for the utterance choice experiments. Optimizing klValueFactor. Softness, non-obedience and alpha are fixed.*

Description

Full-RSA

klValueFactor is optimized.

Softness and non-obedience are set to 0. Alpha is fixed at 1.

Usage

```
RSAModelUttKLDivParamD4(params, data)
```

Arguments

params	One value vector specifying the parameter to be optimized. 1. klValueFactor can be negative, zero or positive: **zero** Don't care about learning about feature preferences of the listener **positive** Care about learning about feature preferences of the listener **negative** Trying to pick non-ambiguous utterances
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.

Details

This function uses RSAModelUttKLDiv_4params.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`RSAModelUttKLDivParamD4A.2` *Cost function for one parameter optimization for the utterance choice experiments. Optimizing klValueFactor. Softness, non-obedience and alpha are fixed.*

Description

Full-RSA

Softness is fixed at 0.2.

non-obedience is fixed at 0.

Alpha is fixed at to 1.

Usage

```
RSAModelUttKLDivParamD4A.2(params, data)
```

Arguments

`params`	One value vector specifying the parameter to be optimized. 1. klValueFactor can be negative, zero or positive: **zero** Don't care about learning about feature preferences of the listener **positive** Care about learning about feature preferences of the listener **negative** Trying to pick non-ambiguous utterances
`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.

Details

This function uses `RSAModelUttKLDiv_4params`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`RSAModelUttKLDivParamD4AB.2` *Cost function for one parameter optimization for the utterance choice experiments. Optimizing klValueFactor. Softness, non-obedience and alpha are fixed.*

Description

Full-RSA

klValueFactor is optimized.

Softness and non-obedience are fixed at 0.2.

Alpha is fixed at 1.

Usage

```
RSAModelUttKLDivParamD4AB.2(params, data)
```

Arguments

`params`	One value vector specifying the parameter to be optimized. 1. klValueFactor can be negative, zero or positive: **zero** Don't care about learning about feature preferences of the listener **positive** Care about learning about feature preferences of the listener **negative** Trying to pick non-ambiguous utterances
`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.

Details

This function uses `RSAModelUttKLDiv_4params`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`RSAModelUttKLDivParamD4B.2` *Cost function for one parameter optimization for the utterance choice experiments. Optimizing klValueFactor. Softness, non-obedience and alpha are fixed.*

Description

Full-RSA

klValueFactor is optimized.

Softness is fixed at 0.

Non-obedience is fixed at 0.2.

Alpha is fixed at 1.

Usage

```
RSAModelUttKLDivParamD4B.2(params, data)
```

Arguments

`params`	One value vector specifying the parameter to be optimized. 1. klValueFactor can be negative, zero or positive: **zero** Don't care about learning about feature preferences of the listener **positive** Care about learning about feature preferences of the listener **negative** Trying to pick non-ambiguous utterances
`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.

Details

This function uses `RSAModelUttKLDiv_4params`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`RSAModelUttKLDiv_3params` *Main cost function for three parameter optimization for the utterance choice experiments. Optimizing softness, non-obedience and klValue-Factor.*

Description

Full-RSA

Actual RSA model Kullback-Leibler divergence determination for the utterance choice experiments.

Softness, non-obedience and klValueFactor are optimized.

Usage

```
RSAModelUttKLDiv_3params(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** A parameter value between `[0,infinity)` (The larger the value 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.

`par2` **non-obedience parameter** 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"*.

`par3` **klValueFactor** A parameter that can be negative, 0 or positive (Here it is set to = 1):

zero Don't care about learning about the feature preferences of the listener

positive Care about learning about the feature preferences of the listener

negative Trying to pick non-ambiguous utterances

Details

This function uses `getBestInfGainUttPreferences`.

And is used by `RSAModelUttKLDivParamA`,

`RSAModelUttKLDivParamB`,

`RSAModelUttKLDivParamC`,

`RSAModelUttKLDivParamBC`,

`RSAModelUttKLDivParamAC`,

`RSAModelUttKLDivParamAB`,

`RSAModelUttKLDivParamABC`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`RSAModelUttKLDiv_4params`

Main cost function for four parameter optimization for the utterance choice experiments. Optimizing softness, non-obedience, alpha and klValueFactor.

Description

Full-RSA

The actual RSA model Kullback-Leibler divergence determination for utterance choice experiments.

Softness, non-obedience, alpha and klValueFactor are optimized.

Usage

```
RSAModelUttKLDiv_4params(data, par1, par2, par3, par4)
```

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** A parameter value between `[0,infinity)` (The larger the value 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.

par2 **non-obedience parameter** 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"*.

par3 **alpha parameter** A parameter value between 0 and 1.
This parameter is an exponential scaling of the speaker choosing the utterance that maximizes the chance of the listener getting the target object right.

par4 klValueFactor can be negative, 0 or positive:

zero Don't care about learning about the feature preferences of the listener

positive Care about learning about the feature preferences of the listener

negative Trying to pick non-ambiguous utterances

Details

This function uses `getBestInfGainUttPreferences`.

And is used by

`RSAModelUttKLDivParamD4`,

`RSAModelUttKLDivParamD4A.2`,

`RSAModelUttKLDivParamD4B.2`,

`RSAModelUttKLDivParamD4AB.2`,

`RSAModelUttKLDivParamAD4`,

`RSAModelUttKLDivParamBD4`,

`RSAModelUttKLDivParamAD4B.2`,

`RSAModelUttKLDivParamBD4A.2`,

`RSAModelUttKLDivParamCD4`,

`RSAModelUttKLDivParamBCD4`,

`RSAModelUttKLDivParamACD4`,

`RSAModelUttKLDivParamABD4`,

`RSAModelUttKLDivParamABCD4`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`simpleBestInfGainUtterance` *Best information gain utterances*

Description

Simple RSA

The ultimate function that determines the utterance preferences of a speaker, who wants to learn about the listener's preferences. The speaker considers all relevant utterances given the currentObjects. He also considers all prior feature value preferences (of the listener) and all possible object choices.

NOTE: This can be manipulated to make the speaker focus on one particular feature type preference by setting the other feature value preferences to zero!

The function infers the resulting posterior feature value preferences of the listener in the particular scenario. It computes the Kullback-Leibler divergence between the expected prior and inferred posterior feature value preferences and finally determines the utility value for the considered utterance in the imagined scenario, adding this utility to all scenarios for each considered utterance.

The utility is determined as the expected information gain between prior and posterior of the determined listener's object preferences.

Usage

```
simpleBestInfGainUtterance(
  preferencesPrior,
  validUtterances,
  currentObjects,
  uttToObjProbs,
  objectPreferenceSoftPriors,
  klValueFactor = 1
)
```

Arguments

`preferencesPrior`
: A vector of the length the validUtterances vector + 1.

It constructed as such:

`preferencesPrior <-rep(1/(length(validUtterances)+1),length(validUtterances)+1).`

The vector contains the probability mass over all feature values present in the scenario plus a "no preference" case.

Gives a prior preferences distribution over the feature values in the scene.

`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`
: 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`).

`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.

`klValueFactor` (here set to = 1) can be negative, 0 or positive:

zero Don't care about learning about feature preferences of the listener

positive Care about learning about feature preferences of the listener

negative Trying to pick non-ambiguous utterances

Value

A vector containing the normalized probability over utterances given the listener's object preference priors.

The utterance with the highest probability is the one that maximizes the information gain for the speaker.

The vector has the same length as the validUtterances vector.

Examples

```
allObjects[currentObjects,]
     shape   pattern  color
[1,] "cloud"  "solid" "blue"
[2,] "circle" "solid" "blue"
[3,] "square" "solid" "blue"

simpleBestInfGainUtterance(preferencesPrior, validUtterances, currentObjects,
uttToObjProbs, objectPreferenceSoftPriors, klValueFactor = 1)

output:
[1] 0.0 0.0 0.0 0.5

Since the all the objects present in the scene are solid and blue,
uttering solid or blue, would be optimal to learn something
about the shape preferences of the listener.
This means the speaker would have the best information gain.
```

`simpleBestInfGainUtteranceWithPrefPriorAll` *Iterative utterance choice function. Utterance preferences of a speaker, who wants to learn about the listener's preferences*

Description

Simple RSA

This function calculates the utility of the utterances. The utterance with the highest utility delivers the best information gain for the speaker about the feature preferences of the listener.

This function is used in the iterative scenarios.

Usage

```
simpleBestInfGainUtteranceWithPrefPriorAll(
  preferencesPriorAll,
  validUtterances,
  currentObjects,
  uttToObjProbs,
  objectPreferenceSoftPriors,
  klValueFactor = 1,
  targetFeature,
  utterancePrior
)
```

Arguments

`preferencesPriorAll`
: A vector of length 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.

`klValueFactor` (here set to = 1) can be negative, 0 or positive:

zero Don't care about learning about feature preferences of the listener

positive Care about learning about feature preferences of the listener

negative Trying to pick non-ambiguous utterances

`targetFeature` A value between 1 and 3, specifying which feature type- color, shape, or pattern- is considered (for preferences).

`utterancePrior` A vector of the same length of the validUtterances vector. It contains zeros.
`utterancePrior <-rep(0,length(validUtterances))`

Details

iterative-version of `simpleBestInfGainUtterance`

Value

posterior preferences over feature values: 3 dimensional array for simulated preferences.

rows: utterances, **columns:** preferences, **blocks**: objects.

It contains the normalized probability over utterances given the listener's object preference priors.

U(utterances | listener's object preference priors).

Examples

```
simpleBestInfGainUtteranceWithPrefPriorAll(preferencesPriorAll,
validUtterances, currentObjects, uttToObjProbs,
objectPreferenceSoftPriors, klValueFactor = 1, targetFeature, utterancePrior)

output:
[[1]]
[1] 0  0  0  0.26  0.088  0.65

[[2]]
, , 1

      [,1]   [,2]  [,3]  [,4] [,5] [,6] [,7] [,8] [,9]
[1,]   0       0     0     0    0    0    0    0    0
[2,]   0       0     0     0    0    0    0    0    0
[3,]   0       0     0     0    0    0    0    0    0
[4,] 0.66   0.0065  0.33   0    0    0    0    0    0
[5,]   0       0     0     0    0    0    0    0    0
[6,] 0.98    0.01   0.01   0    0    0    0    0    0

, , 2

      [,1]   [,2]  [,3]  [,4] [,5] [,6] [,7] [,8] [,9]
[1,]   0       0     0     0    0    0    0    0    0
[2,]   0       0     0     0    0    0    0    0    0
[3,]   0       0     0     0    0    0    0    0    0
[4,] 0.065   0.66   0.33   0    0    0    0    0    0
[5,]   0       0     0     0    0    0    0    0    0
```

```
[6,]  0.01   0.98   0.01   0    0    0    0    0    0

, , 3

     [,1]    [,2]  [,3]  [,4] [,5] [,6] [,7] [,8] [,9]
[1,]   0       0     0     0    0    0    0    0    0
[2,]   0       0     0     0    0    0    0    0    0
[3,]   0       0     0     0    0    0    0    0    0
[4,]   0       0     0     0    0    0    0    0    0
[5,]  0.33   0.33   0.33   0    0    0    0    0    0
[6,] 0.0097 0.0097  0.98   0    0    0    0    0    0
```

`simpleListener` *Simple Listener function*

Description

Simple RSA

The simple listener function determines the listener's object choice given the present objects in the scene and her preferences.

P(obj | utt, listener's object preferences)

Usage

```
simpleListener(utterance, uttToObjProbs, listenerObjectPreferences)
```

Arguments

`utterance`	The uttered word by the speaker that the listener hears. An index referring to one of the values in the vector validUtterances.
`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.
`listenerObjectPreferences`	One of the rows of the 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.

Value

A Vector of length 3. It includes the normalized probability of choosing each of the three objects in the scene, given the utterance and the listener's object preferences.

Examples

```
simpleListener(utterance, uttToObjProbs, listenerObjectPreferences)

output:
[1] 1 0 0
```

`simplePragmaticSpeaker` *Simple pragmatic speaker function*

Description

Simple RSA

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

Usage

```
simplePragmaticSpeaker(
  utterance,
  obj,
  preferencesPrior,
  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.
`preferencesPrior`	A vector of length 9. Probability mass over all feature values present in the scenario plus a "no preference" case. Gives a prior preferences distribution over all (nine) feature values.
`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`	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)`.

`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.

Value

A vector with the same as the validUtterances vector.

Normalized posterior probability over preferences- given the utterance, the object choice by the listener, and prior over preferences of the listener.

Examples

```
simplePragmaticSpeaker(utterance, obj, preferencesPrior,
validUtterances, currentObjects, uttToObjProbs, objectPreferenceSoftPriors)

output:
[1] 0.17 0.17 0.17 0.17 0.17 0.17
```

`simplePragmaticSpeakerWithPrefPriorAll` *Simple pragmatic speaker with all prior preferences*

Description

Simple RSA

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).

Usage

```
simplePragmaticSpeakerWithPrefPriorAll(
  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.

Value

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

Examples

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

output:
[1] 0.12  0.12  0.12  0.12  0.12  0.12 0.12  0.12  0.12
```

`SimpleRSAModelUttKLDivFTFParamA`

Feature Type Focus: Cost function for one parameter optimization for the utterance choice experiments. Optimizing softness. Non-obedience and klValueFactor are fixed.

Description

Simple RSA

Softness is optimized.

Non-obedience is fixed at 0. klValueFactor is fixed at 1.

Usage

```
SimpleRSAModelUttKLDivFTFParamA(par, data)
```

Arguments

`par`	One value vector, which specifies one of three parameters to be optimized: 1. softPrefValue: A parameter value between `[0,infinity)` (The larger the value 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.
`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.

Details

This function uses `SimpleRSAModelUttKLDiv_3paramsFTF`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDivFTFParamA.2`

Feature Type Focus: Cost function for one parameter optimization for the utterance choice experiments. Optimizing softness. Non-obedience and klValueFactor are fixed.

Description

Simple RSA

Softness is optimized.

Non-obedience is fixed at 0.2. klValueFactor is fixed at 1.

Usage

```
SimpleRSAModelUttKLDivFTFParamA.2(par, data)
```

Arguments

`par`	One value vector specifying one of the three parameters to be optimized. 1. softPrefValue: A parameter value between `[0,infinity)` (The larger the value 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.
`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.

Details

This function uses `SimpleRSAModelUttKLDiv_3paramsFTF`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDivFTFParamAB`

Feature Type Focus: Cost function for two parameter optimization for the utterance choice experiments. Optimizing softness and non-obedience. klValueFactor is fixed.

Description

Simple RSA

Softness and non-obedience are optimized.

klValueFactor is fixed at 1.

Usage

```
SimpleRSAModelUttKLDivFTFParamAB(params, data)
```

Arguments

`params`	Two value vector specifying two of the three parameters to be optimized. 1. softPrefValue: A parameter value between `[0,infinity)` (The larger the value 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. 2. non-obedience: This parameter 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"*.
`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.

Details

This function uses `SimpleRSAModelUttKLDiv_3paramsFTF`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDivFTFParamABK`

Feature Type Focus: Cost function for two parameter optimization for the utterance choice experiments. Optimizing softness, non-obedience and klValueFactor.

Description

Simple RSA

Softness, non-obedience and klValueFactor are optimized.

Usage

```
SimpleRSAModelUttKLDivFTFParamABK(params, data)
```

Arguments

`params` Three value vector specifying three of the three parameters to be optimized.

1. softPrefValue: A parameter value between `[0,infinity)` (The larger the value 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.
2. non-obedience:This parameter 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"*.
3. klValueFactor: A parameter that can be negative, 0 or positive:

   zero Don't care about learning about feature preferences of the listener

   positive Care about learning about feature preferences of the listener

   negative Trying to pick non-ambiguous utterances

`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.

Details

This function uses `SimpleRSAModelUttKLDiv_3paramsFTF`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDivFTFParamAK`

Feature Type Focus: Cost function for two parameter optimization for the utterance choice experiments. Optimizing softness and klValue-Factor. Non-obedience is fixed.

Description

Simple RSA

Softness and klValueFacto are optimized.

Non-obedience is fixed at 0.

Usage

```
SimpleRSAModelUttKLDivFTFParamAK(params, data)
```

Arguments

`params`	Two value vector specifying the two parameters to be optimized. 1. softPrefValue: A parameter value between `[0,infinity)` (The larger the value 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. 2. klValueFactor can be negative, zero or positive: **zero** Don't care about learning about feature preferences of the listener **positive** Care about learning about feature preferences of the listener **negative** Trying to pick non-ambiguous utterances
`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.

Details

This function uses `SimpleRSAModelUttKLDiv_3paramsFTF`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDivFTFParamAK.2`

Feature Type Focus: Cost function for two parameter optimization for the utterance choice experiments. Optimizing softness and klValue-Factor. Non-obedience is fixed.

Description

Simple RSA

Softness and klValueFactor are optimized.

Non-obedience is fixed at 0.2.

Usage

```
SimpleRSAModelUttKLDivFTFParamAK.2(params, data)
```

Arguments

`params`	Two value vector specifying the two parameters to be optimized. 1. softPrefValue: A parameter value between `[0,infinity)` (The larger the value 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. 2. klValueFactor: A parameter can be negative, zero or positive: **zero** Don't care about learning about feature preferences of the listener **positive** Care about learning about feature preferences of the listener **negative** Trying to pick non-ambiguous utterances
`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.

Details

This function uses `SimpleRSAModelUttKLDiv_3paramsFTF`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDivFTFParamB`

Feature Type Focus: Cost function for one parameter optimization for the utterance choice experiments. Optimizing non-obedience. Softness and klValueFactor are fixed.

Description

Simple RSA

Non-obedience is optimized.

Softness is fixed at 0. klValueFactor is fixed at 1.

Usage

```
SimpleRSAModelUttKLDivFTFParamB(par, data)
```

Arguments

`par`	One value vector, which specifies one of three parameters to be optimized: 1. non-obedience, This parameter 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"*.
`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.

Details

This function uses `SimpleRSAModelUttKLDiv_3paramsFTF`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDivFTFParamB.2`

Feature Type Focus: Cost function for one parameter optimization for the utterance choice experiments. Optimizing non-obedience. Softness and klValueFactor are fixed.

Description

Simple RSA

Non-obedience is optimized.

Softness is fixed at 0.2. klValueFactor is fixed at 1.

Usage

```
SimpleRSAModelUttKLDivFTFParamB.2(par, data)
```

Arguments

`par`	One value vector specifying one of the three parameters to be optimized. 1. non-obedience: This parameter 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"*.
`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.

Details

This function uses `SimpleRSAModelUttKLDiv_3paramsFTF`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDivFTFParamBK`

Feature Type Focus: Cost function for two parameter optimization for the utterance choice experiments. Optimizing non-obedience and klValueFactor. Softness is fixed.

Description

Simple RSA

Non-obedience and klValueFactor are optimized.

Softness is fixed at 0.

Usage

```
SimpleRSAModelUttKLDivFTFParamBK(params, data)
```

Arguments

`params` Two value vector specifying the two parameters to be optimized.

1. non-obedience: This parameter 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"*.
2. klValueFactor can be negative, zero or positive:
zero Don't care about learning about feature preferences of the listener
positive Care about learning about feature preferences of the listener
negative Trying to pick non-ambiguous utterances

`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.

Details

This function uses `SimpleRSAModelUttKLDiv_3paramsFTF`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDivFTFParamBK.2`

Feature Type Focus: Cost function for two parameter optimization for the utterance choice experiments. Optimizing non-obedience and klValueFactor. Softness is fixed.

Description

Simple RSA

Non-obedience and klValueFactor are optimized.

Softness is fixed at 0.2.

Usage

```
SimpleRSAModelUttKLDivFTFParamBK.2(params, data)
```

Arguments

`params` Two value vector specifying two of the three parameters to be optimized.

1. non-obedience: 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"*.
2. klValueFactor: A parameter that can be negative, 0 or positive:
zero Don't care about learning about feature preferences of the listener
positive Care about learning about feature preferences of the listener
negative Trying to pick non-ambiguous utterances

`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.

Details

This function uses `SimpleRSAModelUttKLDiv_3paramsFTF`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDivFTFParamK` *Feature Type Focus: Cost function for one parameter optimization for the utterance choice experiments. Optimizing klValueFactor. Softness and non-obedience are fixed.*

Description

Simple RSA

klValueFactor is optimized.

Softness and non-obedience are fixed at 0.

Usage

```
SimpleRSAModelUttKLDivFTFParamK(par, data)
```

Arguments

`par`	One value vector, which specifies one of three parameters to be optimized: 1. klValueFactor: A parameter that can be negative, zero or positive: **zero** Don't care about learning about feature preferences of the listener **positive** Care about learning about feature preferences of the listener **negative** Trying to pick non-ambiguous utterances
`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.

Details

This function uses `SimpleRSAModelUttKLDiv_3paramsFTF`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDivFTFParamK.2.0`

Feature Type Focus: Cost function for one parameter optimization for the utterance choice experiments. Optimizing klValueFactor. Softness and non-obedience are fixed.

Description

Simple RSA

klValueFactor is optimized.

Softness is fixed at 0.2. Non-obedience is fixed at 0.

Usage

```
SimpleRSAModelUttKLDivFTFParamK.2.0(par, data)
```

Arguments

`par`	One value vector specifying one of the three parameters to be optimized. 1. klValueFactor: A parameter that can be negative, 0 or positive: **zero** Don't care about learning about feature preferences of the listener **positive** Care about learning about feature preferences of the listener **negative** Trying to pick non-ambiguous utterances
`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.

Details

This function uses `SimpleRSAModelUttKLDiv_3paramsFTF`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDivFTFParamK.2.2`

Feature Type Focus: Cost function for one parameter optimization for the utterance choice experiments. Optimizing klValueFactor. Softness and non-obedience are fixed.

Description

Simple RSA

klValueFactor is optimized.

Softness and non-obedience are fixed at 0.2.

Usage

```
SimpleRSAModelUttKLDivFTFParamK.2.2(par, data)
```

Arguments

`par`	One value vector specifying one of the three parameters to be optimized. 1. klValueFactor: A parameter that can be negative, 0 or positive: **zero** Don't care about learning about feature preferences of the listener **positive** Care about learning about feature preferences of the listener **negative** Trying to pick non-ambiguous utterances
`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.

Details

This function uses `SimpleRSAModelUttKLDiv_3paramsFTF`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDivParamA` *Cost function for one parameter optimization for the utterance choice experiments. Optimizing softness. Non-obedience and klValueFactor are fixed.*

Description

Simple RSA

Softness is optimized. Non-obedience is fixed at 0. klValueFactor is fixed at 1.

Usage

```
SimpleRSAModelUttKLDivParamA(par, data)
```

Arguments

`par`	One value vector, which specifies one of three parameters to be optimized: 1. softPrefValue: A parameter value between `[0,infinity)` (The larger the value 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.
`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.

Details

This is the non-iterative version of `SimpleRSAModelUttKLDivParamA_iterative`.

This function uses `SimpleRSAModelUttKLDiv_3params`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDivParamA.2` *Cost function for one parameter optimization for the utterance choice experiments. Optimizing softness. Non-obedience and klValueFactor are fixed.*

Description

Simple RSA

Softness is optimized.

Non-obedience is fixed at 0.2.

klValueFactor is fixed at 1.

Usage

```
SimpleRSAModelUttKLDivParamA.2(par, data)
```

Arguments

`par`	One value vector specifying one of the three parameters to be optimized. 1. softPrefValue: A parameter value between `[0,infinity)` (The larger the value 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.
`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.

Details

This is the non-iterative version of `SimpleRSAModelUttKLDivParamA.2_iterative`.

This function uses `SimpleRSAModelUttKLDiv_3params`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDivParamA.2_independent`
Cost function for one parameter optimization for the utterance choice experiments. (independent and iterative setting) Optimizing softness. Non-obedience and klValueFactor are fixed.

Description

Simple RSA

Softness is optimized.

Non-obedience is fixed at 0.2. klValueFactor is fixed at 1.

Usage

```
SimpleRSAModelUttKLDivParamA.2_independent(par, data)
```

Arguments

`par`	One value vector specifying one of the three parameters to be optimized. 1. softPrefValue: A parameter value between `[0,infinity)` (The larger the value 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.
`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.

Details

This function uses `SimpleRSAModelUttKLDiv_3params_independent`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDivParamA.2_iterative`
Cost function for one parameter optimization for the utterance choice experiments. (iterative setting) Optimizing softness. Non-obedience and klValueFactor are fixed.

Description

Simple RSA

Softness is optimized.

Non-obedience is fixed at 0.2. klValueFactor is fixed at 1.

Usage

```
SimpleRSAModelUttKLDivParamA.2_iterative(par, data)
```

Arguments

`par`	One value vector specifying one of the three parameters to be optimized. 1. softPrefValue: A parameter value between `[0,infinity)` (The larger the value 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.
`data`	A matrix with data rows. column structure: `[1:OC1,OC2,OC3,4:numUttOptions,6: picked utterance]` **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. **6: picked utterance** The utterance picked by the participant.

Details

This is the iterative version of `SimpleRSAModelUttKLDivParamA.2`.

This function uses `SimpleRSAModelUttKLDiv_3params_iterative`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDivParamAB` *Cost function for two parameter optimization for the utterance choice experiments. Optimizing softness and non-obedience. klValueFactor is fixed.*

Description

Simple RSA

Softness and non-obedience are optimized.

klValueFactor parameter is fixed at 1.

Usage

```
SimpleRSAModelUttKLDivParamAB(params, data)
```

Arguments

`params`	Two value vector specifying two of the three parameters to be optimized. 1. softPrefValue: A parameter value between `[0,infinity)` (The larger the value 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. 2. non-obedience: This parameter 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"*.
`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.

Details

This is the non-iterative version of `SimpleRSAModelUttKLDivParamAB_iterative`.

This function uses `SimpleRSAModelUttKLDiv_3params`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

SimpleRSAModelUttKLDivParamABK *Cost function for three parameter optimization for the utterance choice experiments. Optimizing softness and non-obedience and klValueFactor.*

Description

Simple RSA

Softness, non-obedience and klValueFactor are optimized.

Usage

```
SimpleRSAModelUttKLDivParamABK(params, data)
```

Arguments

params	Three value vector specifying three of the three parameters to be optimized. 1. softPrefValue: A parameter value between `[0,infinity)` (The larger the value 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. 2. non-obedience: This parameter 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"*. 3. klValueFactor: A parameter that can be negative, 0 or positive: **zero** Don't care about learning about feature preferences of the listener **positive** Care about learning about feature preferences of the listener **negative** Trying to pick non-ambiguous utterances
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.

Details

This is the non-iterative version of `SimpleRSAModelUttKLDivParamABK_iterative`.

This function uses `SimpleRSAModelUttKLDiv_3params`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDivParamABK_independent`

Cost function for three parameter optimization for the utterance choice experiments. (independent and iterative setting) Optimizing softness, non-obedience and klValueFactor.

Description

Simple RSA

Softness, non-obedience and klValueFactor are optimized.

Usage

```
SimpleRSAModelUttKLDivParamABK_independent(params, data)
```

Arguments

`params` Three value vector specifying three of the three parameters to be optimized.

1. softPrefValue: A parameter value between `[0,infinity)` (The larger the value 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.
2. non-obedience: This parameter 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"*.
3. klValueFactor: A parameter that can be negative, 0 or positive:
   zero Don't care about learning about feature preferences of the listener
   positive Care about learning about feature preferences of the listener
   negative Trying to pick non-ambiguous utterances

`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.

Details

This function uses `SimpleRSAModelUttKLDiv_3params_independent`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDivParamABK_iterative`
Cost function for three parameter optimization for the utterance choice experiments. (iterative setting) Optimizing softness, non-obedience and klValueFactor.

Description

Simple RSA

Softness, non-obedience and klValueFactor are optimized.

Usage

```
SimpleRSAModelUttKLDivParamABK_iterative(params, data)
```

Arguments

`params` Three value vector specifying three of the three parameters to be optimized.

1. softPrefValue: A parameter value between `[0,infinity)` (The larger the value 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.
2. non-obedience:This parameter 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"*.
3. klValueFactor: A parameter that can be negative, 0 or positive:
   zero Don't care about learning about feature preferences of the listener
   positive Care about learning about feature preferences of the listener
   negative Trying to pick non-ambiguous utterances

`data` A matrix with data rows.
column structure: `[1:OC1,OC2,OC3,4:numUttOptions,6: picked utterance]`
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.
6: picked utterance The utterance picked by the participant.

Details

This is the iterative version of `SimpleRSAModelUttKLDivParamABK`.

This function uses `SimpleRSAModelUttKLDiv_3params_iterative`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDivParamAB_independent`
Cost function for two parameter optimization for the utterance choice experiments. (independent and iterative setting) Optimizing softness and non-obedience. klValueFactor is fixed.

Description

Simple RSA

Softness and non-obedience are optimized.

klValueFactor is fixed at 1.

Usage

```
SimpleRSAModelUttKLDivParamAB_independent(params, data)
```

Arguments

`params` Two value vector specifying two of the three parameters to be optimized.

1. softPrefValue: A parameter value between `[0,infinity)` (The larger the value 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.
2. non-obedience: This parameter 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"*.

`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.

Details

This function uses `SimpleRSAModelUttKLDiv_3params_independent`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDivParamAB_iterative`

Cost function for two parameter optimization for the utterance choice experiments. (iterative setting) Optimizing softness and non-obedience. klValueFactor is fixed.

Description

Simple RSA

Softness and non-obedience are optimized.

klValueFactor is fixed at 1.

Usage

```
SimpleRSAModelUttKLDivParamAB_iterative(params, data)
```

Arguments

`params` Two value vector specifying two of the three parameters to be optimized.

1. softPrefValue: A parameter value between `[0,infinity)` (The larger the value 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.
2. non-obedience: This parameter 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"*.

`data` A matrix with data rows.

column structure: `[1:OC1,OC2,OC3,4:numUttOptions,6: picked utterance]`

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.

6: picked utterance The utterance picked by the participant.

Details

This is the iterative version of SimpleRSAModelUttKLDivParamAB.

This function uses SimpleRSAModelUttKLDiv_3params_iterative.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

SimpleRSAModelUttKLDivParamAK

Cost function for two parameter optimization for the utterance choice experiments. Optimizing Softness and klValueFactor. Non-obedience is fixed.

Description

Simple RSA

Softness and klValueFactor are optimized.

Non-obedience parameter is fixed at 0.

Usage

```
SimpleRSAModelUttKLDivParamAK(params, data)
```

Arguments

params	Two value vector specifying the two parameters to be optimized. 1. softPrefValue: A parameter value between [0,infinity) (The larger the value 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. 2. klValueFactor: A parameter that can be negative, zero or positive: **zero** Don't care about learning about feature preferences of the listener **positive** Care about learning about feature preferences of the listener **negative** Trying to pick non-ambiguous utterances
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.

Details

This is the non-iterative version of `SimpleRSAModelUttKLDivParamAK_iterative`.

This function uses `SimpleRSAModelUttKLDiv_3params`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDivParamAK.2`

Cost function for two parameter optimization for the utterance choice experiments. Optimizing softness and klValueFactor. Non-obedience is fixed.

Description

Simple RSA

Optimizing softness and klValueFactor.

Non-obedience is fixed at 0.2.

Usage

```
SimpleRSAModelUttKLDivParamAK.2(params, data)
```

Arguments

`params` Two value vector specifying the two parameters to be optimized.

1. softPrefValue: A parameter value between `[0,infinity)` (The larger the value 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.
2. klValueFactor: A parameter can be negative, zero or positive:

   zero Don't care about learning about feature preferences of the listener

   positive Care about learning about feature preferences of the listener

   negative Trying to pick non-ambiguous utterances

`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.

Details

This is the non-iterative version of `SimpleRSAModelUttKLDivParamAK.2_iterative`.

This function uses `SimpleRSAModelUttKLDiv_3params`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDivParamAK.2_independent`
Cost function for two parameter optimization for the utterance choice experiments. (independent and iterative setting) Optimizing softness and klValueFactor. Non-obedience is fixed.

Description

Simple RSA

Softness and klValueFactor are optimized.

Non-obedience is fixed at 0.2.

Usage

```
SimpleRSAModelUttKLDivParamAK.2_independent(params, data)
```

Arguments

`params`	Two value vector specifying the two parameters to be optimized. 1. softPrefValue: A parameter value between `[0,infinity)` (The larger the value 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. 2. klValueFactor: A parameter that can be negative, zero or positive: **zero** Don't care about learning about feature preferences of the listener **positive** Care about learning about feature preferences of the listener **negative** Trying to pick non-ambiguous utterances
`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.

Details

This function uses `SimpleRSAModelUttKLDiv_3params_independent`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDivParamAK.2_iterative`

Cost function for two parameter optimization for the utterance choice experiments. (iterative setting) Optimizing softness and klValueFactor. Non-obedience is fixed.

Description

Simple RSA

Softness and klValueFactor are optimized.

Non-obedience is fixed at 0.2.

Usage

```
SimpleRSAModelUttKLDivParamAK.2_iterative(params, data)
```

Arguments

`params`	Two value vector specifying the two parameters to be optimized. 1. softPrefValue: A parameter value between `[0,infinity)` (The larger the value 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. 2. klValueFactor can be negative, zero or positive: **zero** Don't care about learning about feature preferences of the listener **positive** Care about learning about feature preferences of the listener **negative** Trying to pick non-ambiguous utterances
`data`	A matrix with data rows. column structure: `[1:OC1,OC2,OC3,4:numUttOptions,6: picked utterance]` **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. **6: picked utterance** The utterance picked by the participant.

Details

This is the iterative version of `SimpleRSAModelUttKLDivParamAK.2`.

This function uses `SimpleRSAModelUttKLDiv_3params_iterative`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDivParamAK_independent` *Cost function for two parameter optimization for the utterance choice experiments. (independent and iterative setting) Optimizing softness and klValueFactor. Non-obedience is fixed.*

Description

Simple RSA

Softness and klValueFactor are optimized.

Non-obedience is fixed at 0.

Usage

```
SimpleRSAModelUttKLDivParamAK_independent(params, data)
```

Arguments

`params`	Two value vector specifying the two parameters to be optimized. 1. softPrefValue: A parameter value between `[0,infinity)` (The larger the value 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. 2. klValueFactor: A parameter that can be negative, zero or positive: **zero** Don't care about learning about feature preferences of the listener **positive** Care about learning about feature preferences of the listener **negative** Trying to pick non-ambiguous utterances
`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.

Details

This function uses `SimpleRSAModelUttKLDiv_3params_independent`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDivParamAK_iterative` *Cost function for two parameter optimization for the utterance choice experiments. (iterative setting) Optimizing softness and klValueFactor. Non-obedience is fixed.*

Description

Simple RSA

Softness and klValueFactor are optimized.

Non-obedience is fixed at 0.

Usage

```
SimpleRSAModelUttKLDivParamAK_iterative(params, data)
```

Arguments

`params`	Two value vector specifying the two parameters to be optimized. 1. softPrefValue, i.e., the strength of preferring one entity over others. 2. klValueFactor can be negative, zero or positive: **zero** Don't care about learning about feature preferences of the listener **positive** Care about learning about feature preferences of the listener **negative** Trying to pick non-ambiguous utterances
`data`	A matrix with data rows. column structure: `[1:OC1,OC2,OC3,4:numUttOptions,6: picked utterance]` **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. **6: picked utterance** The utterance picked by the participant.

Details

This is the iterative version of `SimpleRSAModelUttKLDivParamAK`

This function uses `SimpleRSAModelUttKLDiv_3params_iterative`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDivParamA_independent` *Cost function for one parameter optimization for the utterance choice experiments. (independent and iterative setting) Optimizing softness. Non-obedience and klValueFactor are fixed.*

Description

Simple RSA

Softness is optimized.

Non-obedience is fixed at 0. klValueFactor is fixed at 1.

Usage

```
SimpleRSAModelUttKLDivParamA_independent(par, data)
```

Arguments

`par`	One value vector, which specifies one of three parameters to be optimized: 1. softPrefValue: A parameter value between `[0,infinity)` (The larger the value 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.
`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.

Details

This function uses `SimpleRSAModelUttKLDiv_3params_independent`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDivParamA_iterative` *Cost function for one parameter optimization for the utterance choice experiments. (iterative setting) Optimizing softness. Non-obedience and klValueFactor are fixed.*

Description

Simple RSA

Softness is optimized.

Non-obedience is fixed at 0. klValueFactor is fixed at 1.

Usage

```
SimpleRSAModelUttKLDivParamA_iterative(par, data)
```

Arguments

`par`	One value vector, which specifies one of three parameters to be optimized: 1. softPrefValue: A parameter value between `[0,infinity)` (The larger the value 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.
`data`	A matrix with data rows. column structure: `[1:OC1,OC2,OC3,4:numUttOptions,6: picked utterance]` **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. **6: picked utterance** The utterance picked by the participant.

Details

This is the iterative version of `SimpleRSAModelUttKLDivParamA`.

This function uses `SimpleRSAModelUttKLDiv_3params_iterative`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDivParamB` *Cost function for one parameter optimization for the utterance choice experiments. Optimizing non-obedience. Non-obedience and klValue-Factor are fixed.*

Description

Simple RSA

Non-obedience is optimized.

softness parameter is fixed at 0. klValueFactor is fixed at 1.

Usage

```
SimpleRSAModelUttKLDivParamB(par, data)
```

Arguments

`par`	One value vector, which specifies one of three parameters to be optimized: 1. non-obedience: This parameter 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"*.
`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.

Details

This is the non-iterative version of `SimpleRSAModelUttKLDivParamB_iterative`.

This function uses `SimpleRSAModelUttKLDiv_3params`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDivParamB.2` *Cost function for one parameter optimization for the utterance choice experiments. Optimizing non-obedience. Softness and klValueFactor are fixed.*

Description

Simple RSA

Non-obedience is optimized.

Softness is fixed at 0.2.

klValueFactor is fixed at 1.

Usage

```
SimpleRSAModelUttKLDivParamB.2(par, data)
```

Arguments

`par`	One value vector specifying one of the three parameters to be optimized. 1. non-obedience: This parameter 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"*.
`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.

Details

This is the non-iterative version of `SimpleRSAModelUttKLDivParamB.2_iterative`

This function uses `SimpleRSAModelUttKLDiv_3params`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDivParamB.2_independent`
Cost function for one parameter optimization for the utterance choice experiments. (independent and iterative setting) Optimizing non-obedience. Softness and klValueFactor are fixed.

Description

Simple RSA

Non-obedience is optimized.

Softness is fixed at 0.2. klValueFactor is fixed at 1.

Usage

```
SimpleRSAModelUttKLDivParamB.2_independent(par, data)
```

Arguments

`par`	One value vector specifying one of the three parameters to be optimized. 1. non-obedience: This parameter 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"*.
`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.

Details

This function uses `SimpleRSAModelUttKLDiv_3params_independent`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDivParamB.2_iterative`
Cost function for one parameter optimization for the utterance choice experiments. (iterative setting) Optimizing non-obedience. Softness and klValueFactor are fixed.

Description

Simple RSA

Non-obedience is optimized.

Softness is fixed at 0.2 klValueFactor is fixed at 1.

Usage

```
SimpleRSAModelUttKLDivParamB.2_iterative(par, data)
```

Arguments

`par`	One value vector specifying one of the three parameters to be optimized. 1. non-obedience: This parameter 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"*.
`data`	A matrix with data rows. column structure: `[1:OC1,OC2,OC3,4:numUttOptions,6: picked utterance]` **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. **6: picked utterance** The utterance picked by the participant.

Details

This is the iterative version of `SimpleRSAModelUttKLDivParamB.2`

This function uses `SimpleRSAModelUttKLDiv_3params_iterative`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDivParamBK` *Cost function for two parameter optimization for the utterance choice experiments. Optimizing non-obedience and klValueFactor. Softness is fixed.*

Description

Simple RSA

Non-obedience and klValueFactor are optimized.

Softness parameter is fixed at 0.

Usage

```
SimpleRSAModelUttKLDivParamBK(params, data)
```

Arguments

`params` Two value vector specifying the two parameters to be optimized.

1. non-obedience: This parameter 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"*.
2. klValueFactor: A parameter that can be negative, zero or positive:
zero Don't care about learning about feature preferences of the listener
positive Care about learning about feature preferences of the listener
negative Trying to pick non-ambiguous utterances

`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.

Details

This is the non-iterative version of `SimpleRSAModelUttKLDivParamBK_iterative`.

This function uses `SimpleRSAModelUttKLDiv_3params`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDivParamBK.2`
Cost function for two parameter optimization for the utterance choice experiments. Optimizing non-obedience and klValueFactor. Softness is fixed.

Description

Simple RSA

Non-obedience and klValueFactor are optimized. Softness is fixed at 0.2.

Usage

```
SimpleRSAModelUttKLDivParamBK.2(params, data)
```

Arguments

`params` Two value vector specifying two of the three parameters to be optimized.

1. non-obedience: 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"*.
2. klValueFactor: A parameter that can be negative, 0 or positive:
zero Don't care about learning about feature preferences of the listener
positive Care about learning about feature preferences of the listener
negative Trying to pick non-ambiguous utterances

`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.

Details

This is the non-iterative version of `SimpleRSAModelUttKLDivParamBK.2_iterative`.

This function uses `SimpleRSAModelUttKLDiv_3params`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDivParamBK.2_independent` *Cost function for two parameter optimization for the utterance choice experiments. (independent and iterative setting) Optimizing non-obedience and klValueFactor. Softness is fixed.*

Description

Simple RSA

Non-obedience and klValueFactor are optimized.

Softness is fixed at 0.2.

Usage

```
SimpleRSAModelUttKLDivParamBK.2_independent(params, data)
```

Arguments

`params`	Two value vector specifying two of the three parameters to be optimized. 1. non-obedience: 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"*. 2. klValueFactor: A parameter that can be negative, 0 or positive: **zero** Don't care about learning about feature preferences of the listener **positive** Care about learning about feature preferences of the listener **negative** Trying to pick non-ambiguous utterances
`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.

Details

This function uses `SimpleRSAModelUttKLDiv_3params_independent`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDivParamBK.2_iterative`
Cost function for two parameter optimization for the utterance choice experiments. (iterative setting) Optimizing non-obedience and klValueFactor. Softness is fixed.

Description

Simple RSA

Non-obedience and klValueFactor are optimized.

Softness is fixed at 0.2.

Usage

```
SimpleRSAModelUttKLDivParamBK.2_iterative(params, data)
```

Arguments

`params` Two value vector specifying two of the three parameters to be optimized.

1. non-obedience: 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"*.
2. klValueFactor: A parameter that can be negative, 0 or positive:
zero Don't care about learning about feature preferences of the listener
positive Care about learning about feature preferences of the listener
negative Trying to pick non-ambiguous utterances

`data` A matrix with data rows.
column structure: `[1:OC1,OC2,OC3,4:numUttOptions,6: picked utterance]`
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.
6: picked utterance The utterance picked by the participant.

Details

This is the iterative version of `SimpleRSAModelUttKLDivParamBK.2`.

This function uses `SimpleRSAModelUttKLDiv_3params_iterative`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDivParamBK_independent` *Cost function for two parameter optimization for the utterance choice experiments. (independent and iterative setting) Optimizing non-obedience and klValueFactor. Softness is fixed.*

Description

Simple RSA

Non-obedience and klValueFactor is optimized.

Softness is fixed at 0.

Usage

```
SimpleRSAModelUttKLDivParamBK_independent(params, data)
```

Arguments

`params`	Two value vector specifying the two parameters to be optimized. 1. non-obedience: This parameter 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"*. 2. klValueFactor: A parameter that can be negative, zero or positive: **zero** Don't care about learning about feature preferences of the listener **positive** Care about learning about feature preferences of the listener **negative** Trying to pick non-ambiguous utterances
`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.

Details

This function uses `SimpleRSAModelUttKLDiv_3params_independent`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDivParamBK_iterative`
Cost function for two parameter optimization for the utterance choice experiments. (iterative setting) Optimizing non-obedience and klValueFactor. Softness is fixed.

Description

Simple RSA

Non-obedience and klValueFactor are optimized.

Softness is fixed at 0.

Usage

```
SimpleRSAModelUttKLDivParamBK_iterative(params, data)
```

Arguments

`params` Two value vector specifying the two parameters to be optimized.

1. non-obedience: This parameter 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"*.
2. klValueFactor can be negative, zero or positive:
zero Don't care about learning about feature preferences of the listener
positive Care about learning about feature preferences of the listener
negative Trying to pick non-ambiguous utterances

`data` A matrix with data rows.
column structure: `[1:OC1,OC2,OC3,4:numUttOptions,6: picked utterance]`
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.
6: picked utterance The utterance picked by the participant.

Details

This is the iterative version of `SimpleRSAModelUttKLDivParamBK`

This function uses `SimpleRSAModelUttKLDiv_3params_iterative`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDivParamB_independent` *Cost function for one parameter optimization for the utterance choice experiments. (independent and iterative setting) Optimizing non-obedience. Softness and klValueFactor are fixed.*

Description

Simple RSA

Non-obedience is optimized.

Softness is fixed at 0. klValueFactor is fixed at 1.

Usage

```
SimpleRSAModelUttKLDivParamB_independent(par, data)
```

Arguments

`par`	One value vector, which specifies one of three parameters to be optimized: 1. non-obedience: This parameter 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"*.
`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.

Details

This function uses `SimpleRSAModelUttKLDiv_3params_independent`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDivParamB_iterative`
Cost function for one parameter optimization for the utterance choice experiments. (iterative setting) Optimizing non-obedience. Softness and klValueFactor are fixed.

Description

Simple RSA

Non-obedience is optimized.

Softness is fixed at 0. klValueFactor is fixed at 1.

Usage

```
SimpleRSAModelUttKLDivParamB_iterative(par, data)
```

Arguments

`par`	One value vector, which specifies one of three parameters to be optimized: 1. non-obedience, This parameter 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"*.
`data`	A matrix with data rows. column structure: `[1:OC1,OC2,OC3,4:numUttOptions,6: picked utterance]` **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. **6: picked utterance** The utterance picked by the participant.

Details

This is the iterative version of `SimpleRSAModelUttKLDivParamB`

This function uses `SimpleRSAModelUttKLDiv_3params_iterative`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDivParamK` *Cost function for one parameter optimization for the utterance choice experiments. Optimizing klValueFactor. Softness and non-obedience are fixed.*

Description

Simple RSA

klValueFactor is optimized.

Softness and non-obedience parameter are fixed at 0.

Usage

```
SimpleRSAModelUttKLDivParamK(par, data)
```

Arguments

`par`	One value vector, which specifies one of three parameters to be optimized: 1. klValueFactor: A parameter that can be negative, zero or positive: **zero** Don't care about learning about feature preferences of the listener **positive** Care about learning about feature preferences of the listener **negative** Trying to pick non-ambiguous utterances
`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.

Details

This is the non-iterative version of `SimpleRSAModelUttKLDivParamK_iterative`.

This function uses `SimpleRSAModelUttKLDiv_3params`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDivParamK.2.0` *Cost function for one parameter optimization for the utterance choice experiments. Optimizing klValueFactor. Softness and non-obedience are fixed.*

Description

Simple RSA

klValueFactor is optimized.

Softness is fixed at 0.2. Non-obedience is fixed at 0.

Usage

```
SimpleRSAModelUttKLDivParamK.2.0(par, data)
```

Arguments

`par`	One value vector specifying one of the three parameters to be optimized. 1. klValueFactor: A parameter that can be negative, 0 or positive: **zero** Don't care about learning about feature preferences of the listener **positive** Care about learning about feature preferences of the listener **negative** Trying to pick non-ambiguous utterances
`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.

Details

This is the non-iterative version of `SimpleRSAModelUttKLDivParamK.2.0_iterative`.

This function uses `SimpleRSAModelUttKLDiv_3params`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDivParamK.2.0_independent`
Cost function for one parameter optimization for the utterance choice experiments. (independent and iterative setting) Optimizing klValue-Factor. Softness and non-obedience are fixed.

Description

Simple RSA

klValueFactor is optimized.

Softness is fixed at 0.2. Non-obedience is fixed at 0.

Usage

```
SimpleRSAModelUttKLDivParamK.2.0_independent(par, data)
```

Arguments

`par`	One value vector specifying one of the three parameters to be optimized: 1. klValueFactor: A parameter that can be negative, 0 or positive: **zero** Don't care about learning about feature preferences of the listener **positive** Care about learning about feature preferences of the listener **negative** Trying to pick non-ambiguous utterances
`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.

Details

This function uses `SimpleRSAModelUttKLDiv_3params_independent`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDivParamK.2.0_iterative`
Cost function for one parameter optimization for the utterance choice experiments. (iterative setting) Optimizing klValueFactor. Softness and non-obedience are fixed.

Description

Simple RSA

klValueFactor is optimized.

Softness is fixed at 0.2. Non-obedience is fixed at 0.

Usage

```
SimpleRSAModelUttKLDivParamK.2.0_iterative(par, data)
```

Arguments

`par`	One value vector specifying one of the three parameters to be optimized. 1. klValueFactor: A parameter that can be negative, 0 or positive: **zero** Don't care about learning about feature preferences of the listener **positive** Care about learning about feature preferences of the listener **negative** Trying to pick non-ambiguous utterances
`data`	A matrix with data rows. column structure: `[1:OC1,OC2,OC3,4:numUttOptions,6: picked utterance]` **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. **6: picked utterance** The utterance picked by the participant.

Details

This is the iterative version of `SimpleRSAModelUttKLDivParamK.2.0`.

This function uses `SimpleRSAModelUttKLDiv_3params_iterative`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDivParamK.2.2` *Cost function for one parameter optimization for the utterance choice experiments. Optimizing klValueFactor. Softness and non-obedience are fixed.*

Description

Simple RSA

klValueFactor is optimized.

Softness and klValueFactor are fixed at 0.2.

Usage

```
SimpleRSAModelUttKLDivParamK.2.2(par, data)
```

Arguments

`par`	One value vector specifying one of the three parameters to be optimized. 1. klValueFactor: A parameter that can be negative, 0 or positive: **zero** Don't care about learning about feature preferences of the listener **positive** Care about learning about feature preferences of the listener **negative** Trying to pick non-ambiguous utterances
`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.

Details

This is the non-iterative version of `SimpleRSAModelUttKLDivParamK.2.2_iterative`

This function uses `SimpleRSAModelUttKLDiv_3params`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDivParamK.2.2_independent`
Cost function for one parameter optimization for the utterance choice experiments. (independent and iterative setting) Optimizing klValue-Factor. Softness and non-obedience are fixed.

Description

Simple RSA

klValueFactor is optimized.

Softness and non-obedience are fixed at 0.2.

Usage

```
SimpleRSAModelUttKLDivParamK.2.2_independent(par, data)
```

Arguments

`par`	One value vector specifying one of the three parameters to be optimized. 1. klValueFactor: A parameter that can be negative, 0 or positive: **zero** Don't care about learning about feature preferences of the listener **positive** Care about learning about feature preferences of the listener **negative** Trying to pick non-ambiguous utterances
`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.

Details

This function uses `SimpleRSAModelUttKLDiv_3params_independent`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDivParamK.2.2_iterative` *Cost function for one parameter optimization for the utterance choice experiments. (iterative setting) Optimizing klValueFactor. Softness and non-obedience are fixed.*

Description

Simple RSA

klValueFactor is optimized.

Softness and non-obedience are fixed at 0.2.

Usage

```
SimpleRSAModelUttKLDivParamK.2.2_iterative(par, data)
```

Arguments

`par`	One value vector specifying one of the three parameters to be optimized. 1. klValueFactor: A parameter that can be negative, 0 or positive: **zero** Don't care about learning about feature preferences of the listener **positive** Care about learning about feature preferences of the listener **negative** Trying to pick non-ambiguous utterances
`data`	A matrix with data rows. column structure: `[1:OC1,OC2,OC3,4:numUttOptions,6: picked utterance]` **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. **6: picked utterance** The utterance picked by the participant.

Details

This is the iterative version of `SimpleRSAModelUttKLDivParamK.2.2`

This function uses `SimpleRSAModelUttKLDiv_3params_iterative`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDivParamK_independent` *Cost function for one parameter optimization for the utterance choice experiments. (independent and iterative setting) Optimizing klValue-Factor. Softness and non-obedience are fixed.*

Description

Simple RSA

klValueFactor is optimized.

Softness and non-obedience are fixed at 0.

Usage

```
SimpleRSAModelUttKLDivParamK_independent(par, data)
```

Arguments

`par`	One value vector specifying one of the three parameters to be optimized: 1. klValueFactor: A parameter that can be negative, 0 or positive: **zero** Don't care about learning about feature preferences of the listener **positive** Care about learning about feature preferences of the listener **negative** Trying to pick non-ambiguous utterances
`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.

Details

This function uses `SimpleRSAModelUttKLDiv_3params_independent`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDivParamK_iterative` *Cost function for one parameter optimization for the utterance choice experiments. (iterative setting) Optimizing klValueFactor. Softness and non-obedience are fixed.*

Description

Simple RSA

klValueFactor is optimized.

Softness and non-obedience are fixed at 0.

Usage

```
SimpleRSAModelUttKLDivParamK_iterative(par, data)
```

Arguments

`par`	One value vector, which specifies one of three parameters to be optimized: 1. klValueFactor: A parameter that can be negative, zero or positive: **zero** Don't care about learning about feature preferences of the listener **positive** Care about learning about feature preferences of the listener **negative** Trying to pick non-ambiguous utterances
`data`	A matrix with data rows. column structure: `[1:OC1,OC2,OC3,4:numUttOptions,6: picked utterance]` **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. **6: picked utterance** The utterance picked by the participant.

Details

This is the iterative version of `SimpleRSAModelUttKLDivParamK`.

This function uses `SimpleRSAModelUttKLDiv_3params_iterative`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

SimpleRSAModelUttKLDiv_3params

Main cost function for three parameter optimization for the utterance choice experiments. Optimizing softness, non-obedience and klValue-Factor.

Description

Simple RSA

Actual RSA model Kullback-Leibler divergence determination for the utterance choice experiments.

Softness, non-obedience and klValueFactor are optimized.

Usage

```
SimpleRSAModelUttKLDiv_3params(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** A parameter value between [0,infinity) (The larger the value 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.
par2	**non-obedience parameter** 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"*.
par3	**klValueFactor** A parameter that can be negative, 0 or positive: **zero** Don't care about learning about the feature preferences of the listener **positive** Care about learning about the feature preferences of the listener **negative** Trying to pick non-ambiguous utterances

Details

This function is used in

SimpleRSAModelUttKLDivParamA,

SimpleRSAModelUttKLDivParamB,

SimpleRSAModelUttKLDivParamK,

SimpleRSAModelUttKLDivParamBK,

SimpleRSAModelUttKLDivParamAK,

SimpleRSAModelUttKLDivParamAK.2,

SimpleRSAModelUttKLDivParamA.2,

SimpleRSAModelUttKLDivParamB.2,

SimpleRSAModelUttKLDivParamK.2.2,

SimpleRSAModelUttKLDivParamK.2.0,

SimpleRSAModelUttKLDivParamBK.2,

SimpleRSAModelUttKLDivParamAB,

SimpleRSAModelUttKLDivParamABK.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDiv_3paramsFTF`

FeatureTypeFocus: Main cost function for three parameter optimization for the utterance choice experiments. Optimizing softness, non-obedience and klValueFactor.

Description

Simple RSA

Actual RSA model Kullback-Leibler divergence determination for the utterance choice experiments.

Softness, non-obedience and klValueFactor are optimized.

Usage

```
SimpleRSAModelUttKLDiv_3paramsFTF(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** A parameter value between `[0,infinity)` (The larger the value 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.

par2 **non-obedience parameter** 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"*.

par3 **klValueFactor** A parameter that can be negative, 0 or positive:

zero Don't care about learning about the feature preferences of the listener

positive Care about learning about the feature preferences of the listener

negative Trying to pick non-ambiguous utterances

Details

This function uses `getSimpleBestInfGainUttPreferencesFTF`.

This function is used in `SimpleRSAModelUttKLDivFTFParamA`,

`SimpleRSAModelUttKLDivFTFParamB`,

`SimpleRSAModelUttKLDivFTFParamK`,

`SimpleRSAModelUttKLDivFTFParamBK`,

`SimpleRSAModelUttKLDivFTFParamAK`,

`SimpleRSAModelUttKLDivFTFParamAK.2`,

`SimpleRSAModelUttKLDivFTFParamA.2`,

`SimpleRSAModelUttKLDivFTFParamB.2`,

`SimpleRSAModelUttKLDivFTFParamK.2.2`,

`SimpleRSAModelUttKLDivFTFParamK.2.0`,

`SimpleRSAModelUttKLDivFTFParamBK.2`,

`SimpleRSAModelUttKLDivFTFParamAK.2`,

`SimpleRSAModelUttKLDivFTFParamAB`,

`SimpleRSAModelUttKLDivFTFParamABK`.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDiv_3params_independent`

Main cost function for three parameter optimization for the utterance choice experiments. Optimizing softness, non-obedience and klValue-Factor. (independent and iterative setting)

Description

Simple RSA

Actual RSA model Kullback-Leibler divergence determination for utterance choice experiments.

Softness, non-obedience and klValueFactor are optimized.

Usage

```
SimpleRSAModelUttKLDiv_3params_independent(data, par1, par2, par3)
```

Arguments

`data`	A matrix with data rows. column structure: `[1:OC1,OC2,OC3,4:numUttOptions,6: picked utterance]` **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. **6: picked utterance** The utterance picked by the participant.
`par1`	**softness parameter** Specifies how much actual feature priorities come into play in the object choice (The larger 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`	**klValueFactor** A parameter that can be negative, 0 or positive: **zero** Don't care about learning about the feature preferences of the listener **positive** Care about learning about the feature preferences of the listener **negative** Trying to pick non-ambiguous utterances

Details

This function uses `getSimpleBestInfGainUttPreferencesIterative`.

This function is used in

`SimpleRSAModelUttKLDivParamA_independent`,

`SimpleRSAModelUttKLDivParamB_independent`,

`SimpleRSAModelUttKLDivParamK_independent`,

`SimpleRSAModelUttKLDivParamBK_independent`,

`SimpleRSAModelUttKLDivParamAK_independent`,

`SimpleRSAModelUttKLDivParamAK.2_independent`,

SimpleRSAModelUttKLDivParamA.2_independent,

SimpleRSAModelUttKLDivParamB.2_independent,

SimpleRSAModelUttKLDivParamK.2.2_independent,

SimpleRSAModelUttKLDivParamK.2.0_independent,

SimpleRSAModelUttKLDivParamBK.2_independent,

SimpleRSAModelUttKLDivParamAB_independent,

SimpleRSAModelUttKLDivParamABK_independent.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

`SimpleRSAModelUttKLDiv_3params_iterative`

Main cost function for three parameter optimization (iterative setting). Optimizing softness, non-obedience and klValueFactor.

Description

Simple RSA

Actual RSA model Kullback-Leibler divergence determination for the utterance choice experiments.

Softness, non-obedience and klValueFactor are optimized.

Usage

```
SimpleRSAModelUttKLDiv_3params_iterative(data, par1, par2, par3)
```

Arguments

`data`	A matrix with data rows. column structure: `[1:OC1,OC2,OC3,4:numUttOptions,6: picked utterance]` **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. **6: picked utterance** The utterance picked by the participant.
`par1`	**softness parameter** A parameter value between `[0,infinity)` (The larger the value 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.

par2 **non-obedience parameter** 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"*.

par3 **klValueFactor** A parameter that can be negative, 0 or positive:

zero Don't care about learning about the feature preferences of the listener

positive Care about learning about the feature preferences of the listener

negative Trying to pick non-ambiguous utterances

Details

This is the iterative version of SimpleRSAModelUttKLDiv_3params.

This function is used in

SimpleRSAModelUttKLDivParamA_iterative,

SimpleRSAModelUttKLDivParamB_iterative,

SimpleRSAModelUttKLDivParamK_iterative,

SimpleRSAModelUttKLDivParamBK_iterative,

SimpleRSAModelUttKLDivParamAK_iterative,

SimpleRSAModelUttKLDivParamAK.2_iterative,

SimpleRSAModelUttKLDivParamA.2_iterative,

SimpleRSAModelUttKLDivParamK.2.2_iterative,

SimpleRSAModelUttKLDivParamK.2.0_iterative,

SimpleRSAModelUttKLDivParamBK.2_iterative,

SimpleRSAModelUttKLDivParamAB_iterative,

SimpleRSAModelUttKLDivParamABK_iterative.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

---

speaker *Speaker function*

Description

Full-RSA

This function determines the utterances that increase the likelihood of the listener picking the object the speaker is referring to.

Usage

```
speaker(obj, listenerObjectPreferences, validUtterances, uttToObjProbs, alpha)
```

Arguments

`obj`	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`).
`listenerObjectPreferences`	One of the rows of the 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.
`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.
`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.
`alpha`	A parameter between 0 and 1. Exponential scaling of the speaker choosing the utterance that maximizes the chance of the listener getting the target object right.

Value

P(utt | obj, listener's object preferences)

A vector of normalized probability for a certain utterance given object and listener's object preferences.

The vector has the same length as the validUtterances vector.

Examples

```
listenerObjectPreferences <- objectPreferenceSoftPriors[[3]]
speaker(obj, listenerObjectPreferences, validUtterances, uttToObjProbs, alpha)

output:
[1] 0.6 0.0 0.0 0.2 0.2
```

`UniformModelKLDiv` *Uniform Model Kullback-Leibler divergence determination*

Description

Full-RSA

Divergence of the observed distribution from the uniform distribution over preferences.

Usage

```
UniformModelKLDiv(data)
```

Arguments

`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.

Value

Minimized Kullback-Leibler divergence and the optimal parameters.

X3_rawData *Experiment 2 raw data (epistemic utterance choice)*

Description

This file contains the raw data from the utterance choice experiment.

Usage

```
X3_rawData
```

Format

A data frame with the following columns:

workerid Participant's ID

item Unique ID of a stimulus

slide_number Indicates trial order.

condition A six-digit code that infers to which category the item belongs. In this experiment, it is only necessary to identify classes that have potentially ambiguous utterances.

language Participant self-reported native language

pref1-pref9 Utterance 1-9.

response1- response9 The value to which the slider (1-9) was adjusted by the participant. Indicates how useful an utterance is for finding out listener preferences. The corresponding property that they are rating is stored under pref1-9. Can be in range 0-1.

Small numbers mean less useful, larger numbers mean more useful.

obj1, obj2, obj3 A three-digit code referring to the object that the speaker intends to signal by the utterance (obj1), or to one of the other two objects present in the scene (obj2, obj3).

The first digit of the code refers to shape, the second digit to texture, and the third digit to color of the object.

1 Shape: 1 cloud, 2 circle, 3 square

2 Texture: 1 solid, 2 striped, 3 polka-dotted

3 Color: 1 blue, 2 red, 3 green

Example: If the target is a solid blue square the code would be: 311.

ambiguous Marks whether a trial contains any ambiguous utterances.

numFeatures The number of feature values present in a scene.

For example, if we have 3 red objects, all of them solid, 1 square, 1 cloud, 1 circle, then the number of features would be 5.

Value in range 3-9.

X4_rawData *Experiment 1 raw data (inferring preferences)*

Description

This file contains the raw data from prior inference experiment.

Usage

```
X4_rawData
```

Format

A data frame with the following columns:

workerid Participant's ID

item Unique ID of a stimulus

slide_number Indicates trial order.

language Participant self-reported native language

pref1-pref6 The feature preference of the listener ('red things', 'clouds', 'striped things', etc.). It is the corresponding property that the participants are rating.

response1-response6 The value to which the slider (1-6) was adjusted by the participant. The corresponding property that they are rating is stored under pref1-6. Can be in range 0-1. The value indicates how much a listener likes objects with a particular property. Small numbers mean less liked, larger numbers mean property is preferred.

target, obj2, obj3 A three-digit code referring to the object that the speaker intends to signal by the utterance (target), or to one of the other two objects present in the scene (obj2, obj3). The first digit of the code refers to shape, the second digit to texture, and the third digit to color of the object.

1 Shape: 1 cloud, 2 circle, 3 square

2 Texture: 1 solid, 2 striped, 3 polka-dotted

3 Color: 1 blue, 2 red, 3 green

Example: If the target is a solid blue square the code would be: 311.

utterance Speaker's utterance. A speaker chooses among the features present in the scene. All possible utterances include: 'cloud', 'circle', 'square', 'solid', 'striped', 'polka-dotted', 'blue', 'red', 'green'.

itemCode A six-digit code that indicates to which category the item belongs. The categorization takes into account how many objects the utterance could possibly refer to, and what properties the target object shares with other objects in the scene. Each category consists of 3 tuples of 2 digits and refers to one feature, where the first tuple always refers to the uttered feature, other features are ordered from most ambiguous to less ambiguous. The more objects share the feature value of the target object, the more ambiguous the feature is. In the example below feature 1 is color since the utterance is 'blue', feature 2 is pattern since the target object shares its pattern (solid) with two other objects. Feature 3 is shape since the target object shares its shape only with one other object. Feature order determines the order of tuples.

The first digit in each tuple would then denote how many objects share the value of the picked object for the corresponding feature. We also reordered the objects so that the picked object would be the first. In the experiment, the target object could take any place in the sequence.

Example: solid blue square, solid blue circle, solid green square.

Ambiguity class: 213222.

Utterance: 'blue'

Target object: blue solid square

How to read code:

First tuple: 21 (color)

2: two objects (the target object + 1 more object) share color.

1: the target object shares color with the second object

Second tuple: 32 (pattern)

3: three objects share the pattern 'solid'.

2: the target object shared the pattern with 2nd and 3rd objects

Third tuple 22 (shape)

2: two objects (the target object + 1 more object) share shape.

2: the target object shares shape with the third object.

The third object is not a possible competitor for being chosen, since the utterance is 'blue', and the object is green.

If a tuple starts with '3' or '1' the second digit would simply denote whether the other objects have different values for that feature or the same (See the second tuple above).

ambiguous Refers to whether the utterance of the speaker is ambiguous or not.

Example: In a scenario with a blue solid circle, a blue striped square and a red polka-dotted cloud. The utterance 'blue' is ambiguous.

Index