

Package: cdlogit (via r-universe)

September 26, 2026

Title Context-Dependent Discrete Choice Models

Version 1.0.0

Description Implements a class of context-dependent discrete choice models, including the standard random utility maximization (RUM) model, classical and generalized random regret minimization (RRM) models, the random advantage maximization (RAM) model, as well as the Emergent Value (EV) model and the contextual random utility maximization (CRUM) model, which explicitly incorporate measures of context dependence. In addition, pairwise normalization (PN) and range normalization (RN) models are supported. Both fixed-parameter and random-parameter specifications can be estimated, the latter allowing for preference heterogeneity. All models are formulated within a logit framework under the assumption of independently and identically distributed (i.i.d.) Gumbel error terms. The implemented methods are based on Tversky and Simonson (1993) <[doi:10.1287/mnsc.39.10.1179](https://doi.org/10.1287/mnsc.39.10.1179)>, Chorus et al. (2014) <[doi:10.1016/j.jbusres.2014.02.010](https://doi.org/10.1016/j.jbusres.2014.02.010)>, Roederkerk et al. (2011) <[doi:10.1509/jmkr.48.4.767](https://doi.org/10.1509/jmkr.48.4.767)>, Guevara and Fukushi (2016) <[doi:10.1016/j.trb.2016.07.012](https://doi.org/10.1016/j.trb.2016.07.012)>, Landry and Webb (2021) <[doi:10.1016/j.jet.2021.105221](https://doi.org/10.1016/j.jet.2021.105221)>, and Daviet and Webb (2023) <[doi:10.1016/j.jmp.2022.102741](https://doi.org/10.1016/j.jmp.2022.102741)>.

Author Yuan Xi [aut, cre], Michel Meulders [ctb], Martina Vandebroek [ctb]

Maintainer Yuan Xi <yuan.xi@kuleuven.be>

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.1

Imports randtoolbox

Depends R (>= 2.10)

LazyData true

NeedsCompilation no

Repository <https://cran.r-universe.dev>

Date/Publication 2026-07-28 16:30:15 UTC
RemoteUrl <https://github.com/cran/cdlogit>
RemoteRef HEAD
RemoteSha c6473b78cc1bc116476606e559c3329a95443658

Contents

camera	2
cdlogit	3
cdmeasure	6
indvpar	8
InformationCriteria	9
invest	9
LRtest	10
predict.cdlogit	11
summary.cdlogit	12
Index	14

camera	<i>Synthetic stated preference data for camera choices</i>
--------	--

Description

A generated dataset camera containing simulated stated choice data for cameras with different attributes.

The synthetic data include three camera alternatives in each choice set, represented by the brands Sony, Nikon, and Canon. Each camera profile is described by three alternative-specific attributes: camera resolution, battery life, and price. Respondent characteristics age, gender, and income are also included as individual-specific variables. The data contain choices of 200 individuals, each of whom faced 9 choice tasks.

Usage

```
data("camera", package = "cdlogit")
```

Format

- A data frame in long format with the following columns:
- id** Individual ID.
 - set_id** Choice set ID.
 - alt_id** Alternative ID, corresponding to the camera brand (Sony, Nikon, or Canon).
 - Megapixel** Camera resolution measured in megapixels (MP).
 - Battery** Battery life of the camera, measured in minutes (Mins).

Price Camera price in euros (EUR).

Age Respondent age in years.

Gender Respondent gender.

Income Respondent income category (Low, Medium, or High).

choice Choice indicator, where 1 indicates the selected alternative and 0 otherwise.

cdlogit

Fit context-dependent discrete choice models

Description

The `cdlogit` function estimates a class of context-dependent discrete choice models. The function supports several models, including the standard random utility maximization (RUM) model, the classical and generalized random regret minimization (RRM) model, the random advantage maximization (RAM) model, the emergent value (EV) model and contextual random utility maximization (CRUM) model that explicitly incorporate measures of context effects, as well as pairwise normalization (PN) and range normalization (RN) models.

The error term is assumed to follow an i.i.d. Gumbel distribution, leading to logit models. Both fixed and mixed (random) specifications are supported.

The function supports a three-part formula syntax similar to that of the `mlogit` package, which includes:

- alternative specific variables with generic coefficients
- choice set specific or individual specific variables with alternative specific coefficients
- alternative specific variables with alternative specific coefficients.

Usage

```
cdlogit(
  formula,
  data,
  index,
  model,
  context_attr,
  context_effect = c("compromise", "attraction", "similarity"),
  reflevel = NULL,
  rpar = NULL,
  ra_weight = 1,
  norm_attr = NULL,
  num_draws = 40,
  start = NULL,
  seed = NULL,
  optim_method = "L-BFGS-B",
  optim_control = list(maxit = 1e+08),
  ...
)
```

Arguments

<code>formula</code>	A symbolic description of the model to be fitted.
<code>data</code>	A data frame in long format, where each row is an alternative within a choice task. The number of alternatives should be the same across all choice sets.
<code>index</code>	A character vector of length 3 giving the column names that identify the individual ID, the choice task (choice set) ID, and the names of the alternatives, respectively. Within each choice task, multiple rows (alternatives) are expected.
<code>model</code>	A character string specifying the type of model to be estimated: "rum" for the standard random utility maximization model, "crum" for the contextual random utility maximization model, "ev" for the emergent value model, "crrm" for the classical random regret minimization model, "grrm" for the generalized random regret minimization model, "ram" for the random advantage maximization model, "pn1" and "pn2" for two specifications of the pairwise normalization model, and "rn1" and "rn2" for two specifications of the range normalization model.
<code>context_attr</code>	A character vector specifying the attribute column names used to compute context effect measures. All listed attributes must be numeric and coded such that larger values indicate more desirable levels. For attributes where smaller is better (e.g., price), you should add a minus sign in front of the name, e.g. use <code>-price</code> so that higher values represent greater desirability.
<code>context_effect</code>	A character vector specifying which context effects to compute. Possible values are "compromise", "attraction", and "similarity", corresponding to the compromise, attraction (decoy), and similarity effects, respectively. By default, <code>context_effect = c("compromise", "attraction", "similarity")</code> . For the CRUM model, users may specify any subset of these effects. For the EV model, only "compromise" and "attraction" effects can be specified.
<code>reflevel</code>	The reference level for the alternative-specific parameters. The default <code>reflevel</code> is the alternative whose name appears first when sorted alphabetically or numerically.
<code>rpar</code>	A vector whose names are attributes with random parameters and values the distribution: "normal" for normal distribution, "gamma" for gamma distribution. By default, <code>rpar = NULL</code> , meaning the parameters to be estimated are fixed, hence common for all individuals.
<code>ra_weight</code>	Only relevant if <code>model</code> is "ram"; A numeric value specifying the weight assigned to the disadvantage term in the denominator of the relative advantage (RA) measure. By default, <code>ra_weight = 1</code> .
<code>norm_attr</code>	Only relevant if <code>model</code> is one of "pn1", "pn2", "rn1", or "rn2"; A character vector of the names of attributes to be normalized.
<code>num_draws</code>	Only relevant if <code>rpar</code> is not <code>NULL</code> ; numeric; the number of Halton draws for the maximum simulated likelihood estimation for the distribution of the individual-specific parameters.
<code>start</code>	Optional numeric vector of user-supplied starting values. If provided, these values are used for optimization initialization.

seed	Optional integer used to generate reproducible random starting values when start is not provided. If both start and seed are NULL, internally defined default starting values are used.
optim_method	Character string specifying the optimization method passed to <code>optim</code> . The default is "L-BFGS-B". Available options include "Nelder-Mead", "BFGS", "CG", "L-BFGS-B", "SANN", and "Brent".
optim_control	Optional list of control parameters passed to <code>optim</code> via the control argument. The default is <code>list(maxit = 1e8)</code> , while all other optimization settings use the corresponding defaults in <code>optim</code> . See <code>optim</code> for details.
...	additional arguments (currently not used).

Value

An object of class `cdlogit` containing model fit results including the estimated parameters, log-likelihood, the variance-covariance matrix of the estimated parameters, the convergence status, the estimated individual level parameters, and the fitted choice probabilities.

References

- Chorus CG, van Cranenburgh S, Dekker T (2014). "Random Regret Minimization for Consumer Choice Modeling: Assessment of Empirical Evidence." *Journal of Business Research*, **67**(11), 2428–2436. doi:10.1016/j.jbusres.2014.02.010.
- Coorevits H, Mulder LD (2023). *Context Effects in an Investment Setting: A RUM vs RRM Approach*. Master's thesis, KU Leuven, Faculty of Economics and Business.
- Daviet R, Webb R (2023). "A Test of Attribute Normalization via a Double Decoy Effect." *Journal of Mathematical Psychology*, **113**, 102741. doi:10.1016/j.jmp.2022.102741.
- Guevara CA, Fukushima M (2016). "Modeling the Decoy Effect with Context-RUM Models: Diagrammatic Analysis and Empirical Evidence from Route Choice SP and Mode Choice RP Case Studies." *Transportation Research Part B: Methodological*, **93**, 318–337. doi:10.1016/j.trb.2016.07.012.
- Landry P, Webb R (2021). "Pairwise Normalization: A Neuroeconomic Theory of Multi-attribute Choice." *Journal of Economic Theory*, **193**, 105221. doi:10.1016/j.jet.2021.105221.
- McFadden D (1974). *Conditional Logit Analysis of Qualitative Choice Behavior*. Academic Press, New York.
- Rooderkerk RP, Van Heerde HJ, Bijmolt THA (2011). "Incorporating Context Effects into a Choice Model." *Journal of Marketing Research*, **48**(4), 767–780. doi:10.1509/jmkr.48.4.767.
- Tversky A, Simonson I (1993). "Context-Dependent Preferences." *Management Science*, **39**(10), 1179–1189. doi:10.1287/mnsc.39.10.1179.
- Xi Y, Meulders M, Vandebroek M (2026). "Context Dependence in Discrete Choice Models." Submitted.

Examples

```
data(invest, package = "cdlogit")
##### MNL model #####
mnl_model = cdlogit(
  formula = choice ~ return + pos - 1,
```

```

data = invest, model = "rum",
index = c("id", "task", "alt"))
summary(mnl_model)

##### EV model #####
ev_invest_model = cdlogit(
  formula = choice ~ return + pos - 1,
  data = invest, model = "ev", index = c("id", "task", "alt"),
  context_attr = c("return", "pos"),
  context_effect = c("compromise", "attraction"))
summary(ev_invest_model)
predict(ev_invest_model, invest[1:3, 2:5])
ev_invest_model$cdmeasure

##### PN model #####
pn_model = cdlogit(
  formula = choice ~ return + pos - 1,
  data = invest, model = "pn1",
  index = c("id", "task", "alt"),
  norm_attr = c("return", "pos"))
summary(pn_model)

```

cdmeasure

Context Effect Measures

Description

Computes context effect measures for each alternative within a choice set, including compromise, attraction (decoy), and similarity measures used in context-dependent discrete choice models. The function operates directly on choice-set designs and can be used without estimating a model. It can be used to check the presence and magnitude of context effects in the design of a choice experiment.

Measures are only available for models of type "crum" or "ev".

Usage

```

cdmeasure(
  data,
  index,
  model,
  context_attr,
  context_effect = c("compromise", "attraction", "similarity"),
  ...
)

```

Arguments

data	A data frame in long format, where each row is an alternative within a choice task. The number of alternatives should be the same across all choice sets.
------	---

index	A character vector of length 3 giving the column names that identify the individual ID (or experimental block ID), the choice task (choice set) ID, and the names of the alternatives, respectively. For the CRUM measures, attributes are standardized within each individual (or block) before the context effect measures are computed.
model	A character string specifying the type of measures in which model "crum" for the contextual random utility maximization model, "ev" for the emergent value model.
context_attr	A character vector specifying the attribute column names used to compute context effect measures. All listed attributes must be numeric and coded such that larger values indicate more desirable levels. For attributes where smaller is better (e.g., price), you should add a minus sign in front of the name, e.g. use -price so that higher values represent greater desirability.
context_effect	A character vector specifying which context effects to compute. Possible values are "compromise", "attraction", and "similarity", corresponding to the compromise, attraction (decoy), and similarity effects, respectively. By default, context_effect = c("compromise", "attraction", "similarity"). For the CRUM model, users may specify any subset of these effects. For the EV model, only "compromise" and "attraction" effects can be specified.
...	Additional arguments (currently not used).

Value

A data frame equal to data with additional columns containing the context effect measures. Only relevant for models of type "crum" or "ev". For the CRUM model, the appended columns CRUM_compromise, CRUM_attraction, and CRUM_dissimilarity correspond to measures of the compromise, attraction (decoy), and similarity effects, respectively. For the EV model, the appended columns EV_compromise and EV_attraction are measure of the compromise and attraction effects.

Examples

```
## Compute context effect measures for a single block of choice sets
## from the invest dataset.
## The resulting measures can be inspected before model estimation
## to evaluate the context-effect structure of the design.
```

```
crum_measures_invest <- cdmeasure(
  data = invest[1:33, c("id", "task", "alt", "return", "pos")],
  index = c("id", "task", "alt"),
  model = "crum",
  context_attr = c("return", "pos"),
  context_effect = c("compromise", "attraction", "similarity")
)
print(crum_measures_invest)
```

```
ev_measures_invest <- cdmeasure(
  data = invest[1:33, c("id", "task", "alt", "return", "pos")],
  index = c("id", "task", "alt"),
  model = "ev",
```

```

    context_attr = c("return", "pos"),
    context_effect = c("compromise", "attraction")
  )
  print(ev_measures_invest)

```

indvpar

Individual-level parameter estimates for mixed cdlogit models

Description

Returns the individual-specific parameter estimates for mixed cdlogit models. These correspond to the posterior means (i.e., empirical Bayes estimates or conditional mean estimates) and posterior standard deviations of the random parameters for each individual.

Usage

```
indvpar(object, ...)
```

Arguments

object	A fitted model object of class cdlogit.
...	Additional arguments (currently not used).

Value

A data frame containing the individual-level estimates of the random parameters. Each row corresponds to an individual in the estimation sample. The first column is the individual id. Subsequent columns contain the posterior mean and posterior standard deviation of each random parameter.

Examples

```

##### Mixed CRUM model #####
mixed_crum_model = cdlogit(
  formula = choice ~ return + pos - 1,
  data = invest, model = "crum",
  index = c("id", "task", "alt"),
  rpar = c(pos = 'normal', return = "normal"),
  context_attr = c("return", "pos"),
  context_effect = c("compromise", "attraction", "similarity"))
summary(mixed_crum_model)
indvpar(mixed_crum_model)
predict(object = mixed_crum_model,
  newdata = invest[1:3, c("task", "alt", "return", "pos")])

```

InformationCriteria	<i>Information criteria for the cdlogit model</i>
---------------------	---

Description

Calculates the Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) for an object of class `cdlogit`, based on the log-likelihood and the number of parameters.

Usage

```
## S3 method for class 'cdlogit'
AIC(object, ...)

## S3 method for class 'cdlogit'
BIC(object, ...)
```

Arguments

<code>object</code>	a fitted model of class <code>cdlogit</code> .
<code>...</code>	additional arguments (currently not used).

Value

For `AIC.cdlogit`, the AIC value. For `BIC.cdlogit`, the BIC value.

invest	<i>Investment choice data</i>
--------	-------------------------------

Description

A stated preference dataset from the investment choice experiment of Coorevits and Mulder (2023). The dataset contains choices among investment alternatives designed to investigate context effects in decision making.

Respondents were presented with hypothetical investment scenarios in which they had to allocate a fixed amount of EUR 1000 to one of three investment options. Each option was characterized by its expected return and probability of success. If the selected investment was successful, the respondent received EUR 1000 plus the stated return; otherwise, the respondent retained the original EUR 1000.

The dataset contains 174 respondents, each completing 11 choice tasks. Each task consists of three unlabelled investment alternatives.

Usage

```
data("invest", package = "cdlogit")
```

Format

- A data frame in long format with the following columns:
- id** Respondent ID.
 - task** Choice set ID.
 - alt** Alternative ID.
 - return** Potential return from a successful investment (EUR).
 - pos** Probability of investment success (percentage).
 - choice** Choice indicator, where 1 indicates the selected alternative and 0 otherwise.
 - gender** Respondent gender.
 - age** Respondent age in years.
 - finknow** Self-reported financial knowledge level (Basic, Intermediate, or Advanced).

References

Coorevits, H. and Mulder, L. D. (2023). *Context Effects in an Investment Setting: A RUM vs RRM Approach*. Master’s thesis, KU Leuven, Faculty of Economics and Business.

LRtest	<i>Likelihood ratio test for cdlogit models</i>
--------	---

Description

The LRtest function performs a likelihood ratio test for two nested cdlogit models, to check whether one model provides a significantly better fit to the data than the other model.

Usage

LRtest(model1, model2, boundary = FALSE)

Arguments

- model1 A fitted model of class cdlogit.
- model2 A fitted model of class cdlogit.
- boundary Logical. If TRUE, the likelihood ratio statistic is evaluated using a chi-bar-square distribution (i.e., mixture of chi-square distributions). This option is intended for comparisons in which the null hypothesis places the parameters being tested on the boundary of the parameter space. A common example is testing whether the standard deviation of a random coefficient is equal to zero, since standard deviations are constrained to be non-negative. The default is FALSE, in which case the standard chi-square distribution is used.

Value

Summary of the likelihood ratio test, showing degrees of freedom, log-likelihood values, test statistic, and p-value.

References

Self SG, Liang KY (1987). "Asymptotic Properties of Maximum Likelihood Estimators and Likelihood Ratio Tests Under Nonstandard Conditions." *Journal of the American Statistical Association*, **82**(398), 605–610. doi:10.2307/2289471.

Examples

```
### Conduct a likelihood ratio test for nested models ###
data(invest, package = "cdlogit")
mnl_model = cdlogit(
  formula = choice ~ return + pos - 1,
  data = invest, model = "mnl", index = c("id", "task", "alt"))
crum_model = cdlogit(
  formula = choice ~ return + pos - 1,
  data = invest, model = "crum", index = c("id", "task", "alt"),
  context_attr = c("return", "pos"),
  context_effect = c("compromise", "attraction", "similarity"))
LRtest(mnl_model, crum_model)

### Conduct a likelihood ratio test when the parameters
### are on the boundary of the parameter space
## GRRM model
grrm_model = cdlogit(
  formula = choice ~ return + pos - 1,
  data = invest,
  model = "grrm",
  index = c("id", "task", "alt"))
summary(grrm_model)
## CRRM model
crrm_model = cdlogit(
  formula = choice ~ return + pos - 1,
  data = invest,
  model = "crrm",
  index = c("id", "task", "alt"))
summary(crrm_model)
LRtest(grrm_model, crrm_model)
```

predict.cdlogit

Predict choice probabilities from a cdlogit model

Description

Computes predicted choice probabilities for each alternative in (new) choice sets using a fitted cdlogit model.

Usage

```
## S3 method for class 'cdlogit'
predict(object, newdata, ...)
```

Arguments

object	A fitted model object of class <code>cdlogit</code> .
newdata	A data frame containing the choice sets for which predictions should be computed.
...	additional arguments (currently not used).

Value

A data frame equal to `newdata` with additional columns containing predicted choice probabilities. For models with fixed parameters, probabilities are computed directly from the estimated parameters. For mixed models, the function provides:

- `population_pred`: Population-level predicted probabilities obtained by integrating over the estimated distribution of the random coefficients, conditional on the observed characteristics included in the model. These predictions are particularly relevant for new individuals for whom no choice history is available;
- `posterior_pred`: Individual-specific predicted probabilities obtained by conditioning on the observed choices of each individual and integrating over the corresponding posterior distribution of the random coefficients. For a new choice set, this yields predicted probabilities for each individual in the estimation sample;
- `plugin_pred`: Individual-specific predicted probabilities computed by evaluating the choice probabilities at the mean of the individual-specific posterior distribution. This provides a computationally convenient approximation to posterior prediction.

summary.cdlogit

Summary method for cdlogit model objects.

Description

Provides a summary of the estimation results of the fitted `cdlogit` model.

Usage

```
## S3 method for class 'cdlogit'
summary(object, ...)
```

Arguments

object	a fitted model of class <code>cdlogit</code> .
...	additional arguments (currently not used).

Value

Estimation results of the fitted `cdlogit` model, including:

- The model call, showing how the `cdlogit` model was specified.
- A table that includes the estimates of the coefficients, and the corresponding standard errors, z-values, p-values, and significance codes.
- The log-likelihood of the fitted model.

Index

* datasets

camera, [2](#)

invest, [9](#)

AIC.cdlogit (InformationCriteria), [9](#)

BIC.cdlogit (InformationCriteria), [9](#)

camera, [2](#)

cdlogit, [3](#)

cdmeasure, [6](#)

indvpar, [8](#)

InformationCriteria, [9](#)

invest, [9](#)

LRtest, [10](#)

optim, [5](#)

predict.cdlogit, [11](#)

summary.cdlogit, [12](#)