

Package ‘lboxcox’

October 5, 2026

Type Package

Title Implementation of Logistic Box-Cox Regression

Version 2.0.1

Date 2026-09-05

Maintainer Li Xing <jnjfayaa@gmail.com>

Description Implements a logistic Box-Cox model that adds a shape parameter to a routine logistic regression model to flexibly estimate the shape and strength of the relationship between a binary outcome and a continuous predictor, adjusting for covariates and survey weights. This model is fully described in Xing, L. et al. (2021) <[doi:10.1002/cjs.11587](https://doi.org/10.1002/cjs.11587)>. This version extends the original 'lboxcox' package (1.1) with numerically stabilized likelihood/gradient calculations, vectorized data preprocessing, and a bootstrap-ensemble estimator.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Depends R (>= 3.5.0), survey

Imports maxLik, caret, doParallel, foreach, MASS, stats

NeedsCompilation no

Author Li Xing [cre, aut],
Shiyu Xu [aut],
Jing Wang [aut],
Kohlton Booth [aut],
Xuekui Zhang [aut],
Igor Burstyn [aut],
Paul Gustafson [aut]

Repository CRAN

Date/Publication 2026-10-04 23:00:22 UTC

Contents

depress	2
devr	3
lbc_maxlik	3
lbc_train_all	4
lbc_train_bagging	6
lbc_train_ms	7
lboxcox_cv.fit	8
lboxcox_cv.predict	9
lboxcox_maxLik.predict	9
lboxcox_maxLik_el.predict	10
median_effect	11
Index	12

depress	<i>Depression dataset</i>
---------	---------------------------

Description

The depress data frame contains 8,893 adults aged 20 years or older from the 2005-2006 and 2007-2008 National Health and Nutrition Examination Survey (NHANES) cycles.

Usage

depress

Format

Sample survey data with columns:

depression binary response variable equal to 1 for a Patient Health Questionnaire-9 (PHQ-9) score of at least 10 and 0 otherwise

mercury total blood mercury concentration in micrograms per litre

age age of the participant in years

gender 0 if the participant is female and 1 if the participant is male

weight four-year Day 1 dietary sampling weight, formed as WTDRD1 / 2 for the two combined NHANES cycles

Source

National Center for Health Statistics, NHANES 2005-2006 and 2007-2008. The analytic data were adapted from the data used by Xing, L., Zhang, X., Burstyn, I., & Gustafson, P. (2021). On logistic Box-Cox regression for flexibly estimating the shape and strength of exposure-disease relationships. Canadian Journal of Statistics, 49(3), 808-825.

devr	<i>Deviance residual prediction error</i>
------	---

Description

Computes the sum of absolute signed deviance residuals between binary outcomes and predicted probabilities. Used as the cross-validation criterion in [lboxcox_cv.fit](#) and, more generally, as a prediction-error summary for comparing fitted models.

Usage

```
devr(y.bin, y.pred)
```

Arguments

y.bin	binary (0/1) outcome vector.
y.pred	predicted probabilities, same length as y.bin.

Value

a single numeric value: the sum of absolute deviance residuals.

lbc_maxlik	<i>Fit a logistic Box-Cox model by direct likelihood maximization</i>
------------	---

Description

Trains the given formula using a logistic Box-Cox model whose parameters (intercept, slope, Box-Cox shape, and covariate coefficients) are found by maximizing the log-likelihood directly with `maxLik::maxLik`. If the resulting lambda estimate falls outside $[0, 2]$, falls back to the best of a grid of `svyglm` fits at different lambda values (see [lbc_train_ms](#) for a version that always searches the grid).

Usage

```
lbc_maxlik(
  formula,
  weight_column_name,
  data,
  init = NULL,
  svy_lambda_vector = seq(0, 2, length = 4),
  init_lambda_vector = seq(0, 2, length = 100),
  num_cores = 1,
  seed = NULL
)
```

Arguments

formula	a formula of the form $y \sim x + z1 + z2$ where y is a binary response variable, x is a continuous predictor variable, and $z1$, $z2$, ... are covariates.
weight_column_name	the name of the column in data containing the survey weights, a numeric vector of weights, or NULL/1 for unweighted analysis.
data	dataframe containing the dataset to train on.
init	initial estimates for the coefficients. If NULL, a grid of svyglm models is used to build a starting vector.
svy_lambda_vector	values of lambda used in training the svyglm model that supplies initial coefficient estimates. Ignored if init is not NULL.
init_lambda_vector	values of lambda used, as a fallback, to find the best-log-likelihood starting point if the direct maximization does not converge to a lambda in $[0, 2]$.
num_cores	the number of cores used when searching svy_lambda_vector. Ignored if init is not NULL.
seed	deprecated compatibility argument; ignored because the BFGS optimizer is deterministic. Control bootstrap reproducibility by calling <code>set.seed()</code> before <code>lbc_train_bagging()</code> or <code>lbc_train_all()</code> .

Value

object of class `maxLik` from the **maxLik** package (or, on fallback, of class `svyglm`). Contains the coefficient estimates in `$estimate` (named `Beta_0`, `Beta_1`, `Lambda`, then covariates) and a convergence flag in `$error`.

Note

This is reliant on the following work:

Henningsen, A., Toomet, O. (2011). `maxLik`: A package for maximum likelihood estimation in R. *Computational Statistics*, 26(3), 443-458.

<code>lbc_train_all</code>	<i>Bootstrap-ensemble logistic Box-Cox model around <code>lbc_train_ms</code></i>
----------------------------	---

Description

Fits `lbc_train_ms` on each of 100 bootstrap resamples of data in parallel, takes the median of the per-resample lambda estimates, and refits the final model at that lambda on the full data. More expensive than `lbc_train_bagging` (each resample itself searches a lambda grid via `lbc_train_ms`) but more robust to poor starting values within each resample. Bootstrap samples use the caller's current random-number state. Call `set.seed()` before this function when reproducible resamples are required.

Usage

```
lbc_train_all(  
  formula,  
  weight_column_name,  
  data,  
  init = NULL,  
  svy_lambda_vector = seq(0, 2, length = 100),  
  num_cores = 1,  
  cores = 5  
)
```

Arguments

formula	a formula of the form $y \sim x + z1 + z2$ where y is a binary response variable, x is a continuous predictor variable, and $z1$, $z2$, ... are covariates.
weight_column_name	the name of the column in data containing the survey weights, a numeric vector of weights, or NULL/1 for unweighted analysis.
data	dataframe containing the dataset to train on.
init	initial estimates for the coefficients. If NULL, a grid of <code>svyglm</code> models is used to build a starting vector.
svy_lambda_vector	values of λ used to build the grid of starting vectors that <code>maxLik</code> is run from.
num_cores	the number of cores used when searching <code>svy_lambda_vector</code> . Ignored if <code>init</code> is not NULL.
cores	number of cores to parallelize the bootstrap resamples over.

Value

a list of length 3: the final refit model (as returned by `lbc_train_ms`), the list of per-resample fits (NA for any resample where `lbc_train_ms` errored), and the count of resamples that errored.

Note

This is reliant on the following work:

Microsoft Corporation, Weston, S. (2020). `foreach`: Provides Foreach Looping Construct. R package version 1.5.1.

Microsoft Corporation, Weston, S. (2020). `doParallel`: Foreach Parallel Adaptor for the 'parallel' Package. R package version 1.0.16.

lbc_train_bagging	<i>Bootstrap-ensemble logistic Box-Cox model around lbc_maxlik</i>
-------------------	--

Description

Fits `lbc_maxlik` on each of 100 bootstrap resamples of data in parallel, takes the median of the per-resample lambda estimates, and refits the final model at that lambda on the full data. Bootstrap samples use the caller's current random-number state. Call `set.seed()` before this function when reproducible resamples are required.

Usage

```
lbc_train_bagging(
  formula,
  weight_column_name,
  data,
  init = NULL,
  svy_lambda_vector = seq(0, 2, length = 4),
  init_lambda_vector = seq(0, 2, length = 100),
  num_cores = 1,
  cores = 10
)
```

Arguments

<code>formula</code>	a formula of the form $y \sim x + z1 + z2$ where y is a binary response variable, x is a continuous predictor variable, and $z1$, $z2$, ... are covariates.
<code>weight_column_name</code>	the name of the column in data containing the survey weights, a numeric vector of weights, or <code>NULL/1</code> for unweighted analysis.
<code>data</code>	dataframe containing the dataset to train on.
<code>init</code>	initial estimates for the coefficients. If <code>NULL</code> , a grid of <code>svyglm</code> models is used to build a starting vector.
<code>svy_lambda_vector</code>	values of lambda used in training the <code>svyglm</code> model that supplies initial coefficient estimates. Ignored if <code>init</code> is not <code>NULL</code> .
<code>init_lambda_vector</code>	values of lambda used, as a fallback, to find the best-log-likelihood starting point if the direct maximization does not converge to a lambda in $[0, 2]$.
<code>num_cores</code>	the number of cores used when searching <code>svy_lambda_vector</code> . Ignored if <code>init</code> is not <code>NULL</code> .
<code>cores</code>	number of cores to parallelize the bootstrap resamples over.

Value

a list of length 3: the final refit model (as returned by `lbc_maxlik`/its fallback), the list of per-resample fits (NA for any resample where `lbc_maxlik` errored), and the count of resamples that errored.

Note

This is reliant on the following work:

Microsoft Corporation, Weston, S. (2020). `foreach`: Provides Foreach Looping Construct. R package version 1.5.1.

Microsoft Corporation, Weston, S. (2020). `doParallel`: Foreach Parallel Adaptor for the 'parallel' Package. R package version 1.0.16.

lbc_train_ms	<i>Fit a logistic Box-Cox model from multiple starting points</i>
--------------	---

Description

Trains the given formula using a logistic Box-Cox model by running `maxLik::maxLik` from a grid of `svyglm`-based starting vectors (one per entry of `svy_lambda_vector`) and keeping the fit with the highest achieved log-likelihood. More expensive than `lbc_maxlik` but more robust to a poor single starting point.

Usage

```
lbc_train_ms(
  formula,
  weight_column_name,
  data,
  init = NULL,
  svy_lambda_vector = seq(0, 2, length = 100),
  num_cores = 1
)
```

Arguments

formula	a formula of the form $y \sim x + z1 + z2$ where y is a binary response variable, x is a continuous predictor variable, and $z1$, $z2$, ... are covariates.
weight_column_name	the name of the column in data containing the survey weights, a numeric vector of weights, or NULL/1 for unweighted analysis.
data	dataframe containing the dataset to train on.
init	initial estimates for the coefficients. If NULL, a grid of <code>svyglm</code> models is used to build a starting vector.

svy_lambda_vector	values of lambda used to build the grid of starting vectors that maxLik is run from.
num_cores	the number of cores used when searching svy_lambda_vector. Ignored if init is not NULL.

Value

the refit svyglm model at the selected lambda, with \$estimate (named Beta_0, Beta_1, Lambda, then covariates) and a convergence flag in \$error; error = 1 means no continuous BFGS run converged and the exact profile grid was used as a fallback.

Note

This is reliant on the following work:

Henningsen, A., Toomet, O. (2011). maxLik: A package for maximum likelihood estimation in R. Computational Statistics, 26(3), 443-458.

lboxcox_cv.fit	<i>Select the Box-Cox lambda by cross-validated deviance residual</i>
----------------	---

Description

Trains the given formula using a logistic Box-Cox model whose lambda is chosen, instead of by likelihood maximization, as the value in lambda_vector with lowest mean k-fold cross-validated [devr](#) (deviance residual prediction error). The model is then refit on the full data at the selected lambda.

Usage

```
lboxcox_cv.fit(
  mydata,
  ixx,
  iyy,
  formula,
  weight_column_name = NULL,
  lambda_vector = seq(0, 2, length = 100),
  k
)
```

Arguments

mydata	data frame containing the dataset to train on.
ixx	the (untransformed) primary predictor vector, e.g. mydata\$xx.
iyx	the binary outcome vector, e.g. mydata\$yy.
formula	a formula of the form $y \sim x + z1 + z2$ where y is a binary response variable, x is a continuous predictor variable, and z1, z2, ... are covariates.

weight_column_name	the name of the column in mydata containing the survey weights, a numeric vector of weights, or NULL/1 for unweighted analysis.
lambda_vector	grid of Box-Cox lambda values to select from.
k	number of cross-validation folds.

Value

the refit svyglm model at the selected lambda, with \$estimate (named Beta_0, Beta_1, Lambda, then covariates) attached.

lboxcox_cv.predict	<i>Predict from a lboxcox_cv.fit model</i>
--------------------	--

Description

Predict from a lboxcox_cv.fit model

Usage

```
lboxcox_cv.predict(myCVfit, newdata, formula)
```

Arguments

myCVfit	a model fit by lboxcox_cv.fit .
newdata	data frame of new observations to predict on.
formula	the same formula used to fit myCVfit.

Value

a numeric vector of predicted probabilities.

lboxcox_maxLik.predict	<i>Predict from a fitted logistic Box-Cox model</i>
------------------------	---

Description

Produces predicted probabilities for newdata from a model fit by [lbc_maxlik](#), [lbc_train_ms](#), or the first (\$estimate-bearing) element returned by [lbc_train_bagging](#) or [lbc_train_all](#).

Usage

```
lboxcox_maxLik.predict(myMaxLikfit, newdata, formula)
```

Arguments

myMaxLikfit	a fitted model with a named <code>\$estimate</code> vector (Beta_0, Beta_1, Lambda, then covariates).
newdata	data frame of new observations to predict on.
formula	the same formula used to fit myMaxLikfit.

Value

a numeric vector of predicted probabilities.

`lboxcox_maxLik_el.predict`

Predict from a bootstrap-ensemble logistic Box-Cox model

Description

Averages predicted probabilities across the (up to 100) per-resample fits returned as the second element of `lbc_train_bagging` or `lbc_train_all`, skipping resamples that failed to converge.

Usage

```
lboxcox_maxLik_el.predict(myMaxLik_elfit, newdata, formula)
```

Arguments

myMaxLik_elfit	the length-3 list returned by <code>lbc_train_bagging</code> or <code>lbc_train_all</code> .
newdata	data frame of new observations to predict on.
formula	the same formula used to fit myMaxLik_elfit.

Value

a numeric vector of predicted probabilities, averaged over the converged bootstrap resamples.

median_effect	<i>Median exposure-response slope of a fitted logistic Box-Cox model</i>
---------------	--

Description

Calculates a number that represents the overall gradient measurement between the (log-scale) predictor and the log-odds of the outcome, evaluated at the survey-weighted mean of $\log(x)$, along with a Wald-type 95% confidence interval derived from the model's Hessian.

Usage

```
median_effect(formula, weight_column_name, data, trained_model)
```

Arguments

formula	the formula used to train the logistic Box-Cox model.
weight_column_name	the name of the column in data containing the survey weights, a numeric vector of weights, or NULL/1 for unweighted analysis.
data	dataframe containing the dataset the model was trained on.
trained_model	the already-trained model, e.g. the output of <code>lbc_maxlik</code> or <code>lbc_train_ms</code> . Must expose <code>\$estimate</code> (with <code>Beta_1</code> and <code>Lambda</code>). For a <code>svyglm</code> result, the confidence interval is conditional on its selected <code>lambda</code> .

Value

a named numeric vector with elements median effect, lower 95% ci, and upper 95% ci.

Index

* **datasets**

depress, [2](#)

depress, [2](#)

devr, [3](#), [8](#)

lbc_maxlik, [3](#), [6](#), [7](#), [9](#), [11](#)

lbc_train_all, [4](#), [9](#), [10](#)

lbc_train_bagging, [4](#), [6](#), [9](#), [10](#)

lbc_train_ms, [3-5](#), [7](#), [9](#), [11](#)

lboxcox_cv.fit, [3](#), [8](#), [9](#)

lboxcox_cv.predict, [9](#)

lboxcox_maxLik.predict, [9](#)

lboxcox_maxLik_el.predict, [10](#)

median_effect, [11](#)