Package 'glmdisc'

November 14, 2026

Type Package

Title Discretization and Grouping for Logistic Regression

Version 0.8.1

Date 2026-10-23

Maintainer Adrien Ehrhardt <adrien.ehrhardt@polytechnique.edu>

Description A Stochastic-Expectation-Maximization (SEM) algorithm (Celeux et al. (1995) <https://inria.hal.science/inria-00074164>) associated with a Gibbs sampler which purpose is to learn a constrained representation for logistic regression that is called quantization (Ehrhardt et al. (2019) <doi:10.48550/arXiv.1903.08920>). Continuous features are discretized and categorical features' values are grouped to produce a better logistic regression model. Pairwise interactions between the discretized and grouped features are dynamically added to the model through a Metropolis-Hastings algorithm (Hastings, W. K. (1970) <doi:10.1093/biomet/57.1.97>).

License GPL (>= 2)

Encoding UTF-8

Language en-US

Config/testthat/edition 3

Config/Needs/website pkgdown

Imports gam, graphics, MASS, methods, Rcpp (>= 1.0.7), RcppNumerical, stats

LinkingTo Rcpp, RcppEigen, RcppNumerical

URL https://adimajo.github.io/glmdisc/, https://github.com/adimajo/glmdisc

BugReports https://github.com/adimajo/glmdisc/issues

Suggests covr, DT, knitr, nnet, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Collate 'RcppExports.R' 'allClasses.R' 'cut.dataset.R' 'discretize.link.R' 'features.R' 'generic_cutpoints.R' 'generic_discretize.R' 'glmdisc.R' 'initialization.R'

'link_models.R' 'logit_codes.R' 'method_cutpoints.R'
'method_discretize.R' 'method_plot.R' 'method_predict.R'
'methods_disc.R' 'normalizedGini.R' 'semDiscretization.R'
'sem_engine.R' 'utils.R'

Config/roxygen2/version 8.1.1

NeedsCompilation yes

Author Adrien Ehrhardt [aut, cre] (ORCID:
<https://orcid.org/0000-0002-4448-3644>),
Vincent Vandewalle [aut] (ORCID:
<https://orcid.org/0000-0003-2946-9059>),
Christophe Biernacki [ctb] (ORCID:
<https://orcid.org/0000-0002-8995-0568>),
Philippe Heinrich [ctb]

Repository CRAN

Date/Publication 2026-11-14 08:20:15 UTC

Contents

glmdisc-package 2
cutpoints 3
discretize 4
glmdisc 5
glmdisc-class 9
normalizedGini 9
plot 10
predict 11
predictlogisticRegression 12
print.glmdisc 13
summary.glmdisc 13

Index **15**

`glmdisc-package` *glmdisc: A package for discretizing continuous features, grouping categorical features' values and optimizing it for logistic regression*

Description

The glmdisc package provides two important functions: glmdisc and its associate method discretize.

`glmdisc` function

The `glmdisc` function discretizes a training set using an SEM-Gibbs based method.

`discretize` function

The `discretize` function will discretize a new input dataset given a discretization scheme of S4 class `glmdisc`.

`cutpoints` function

The `cutpoints` function will provide the cutpoints / groupings of a discretization scheme of S4 class `glmdisc` in a list.

`predict` function

The `predict` function will discretize a raw test set, given a provided discretization scheme of S4 class `glmdisc`, using the `discretize` function and return the predicted probabilities.

Miscellaneous

We provide as well the classical `show`, `print`, `summary` functions, as well as `normalizedGini` that is used to calculate the Gini index (a classical Credit Scoring goodness-of-fit indicator).

Author(s)

Adrien Ehrhardt.

See Also

Useful links:

- `https://adimajo.github.io`
- Report bugs at `https://github.com/adimajo/glmdisc/issues`

`cutpoints` *Obtaining the cutpoints and / or groupings of a discretization*

Description

This defines the generic method "cutpoints" which will provide the cutpoints of a discretization scheme of S4 class `glmdisc`.

This defines the method to provide the cutpoints of a trained glmdisc.

Usage

```
cutpoints(object)

## S4 method for signature 'glmdisc'
cutpoints(object)
```

Arguments

object	generic glmdisc object
glmdisc	The trained glmdisc S4 object.

Value

A named list with one element per feature. For a numeric feature, a numeric vector with the cut-points between consecutive levels of the discretization (empty if the feature has a single level). For a factor, a named integer vector that gives, for each level of the factor, the number of the group it was assigned to.

Author(s)

Adrien Ehrhardt.

Examples

```
# Simulation of a discretized logit model
set.seed(1)
x <- matrix(runif(300), nrow = 100, ncol = 3)
cuts <- seq(0, 1, length.out = 4)
xd <- apply(x, 2, function(col) as.numeric(cut(col, cuts)))
theta <- t(matrix(c(0, 0, 0, 2, 2, 2, -2, -2, -2), ncol = 3, nrow = 3))
log_odd <- rowSums(t(sapply(seq_along(xd[, 1]), function(row_id) {
  sapply(
    seq_along(xd[row_id, ]),
    function(element) theta[xd[row_id, element], element]
  )
})))
y <- rbinom(100, 1, 1 / (1 + exp(-log_odd)))

sem_disc <- glmdisc(x, y,
  iter = 50, m_start = 4, test = FALSE,
  validation = FALSE, criterion = "aic"
)
cutpoints(sem_disc)
```

---

discretize *Prediction on a raw test set of the best logistic regression model on discretized / grouped data*

Description

This function discretizes a user-provided test dataset given a discretization scheme provided by an S4 glmdisc object. It then applies the learnt logistic regression model and outputs its prediction (see predict.glm).

This defines the method "discretize" which will discretize a new input dataset given a discretization scheme of S4 class glmdisc

Usage

```
discretize(object, data)

## S4 method for signature 'glmdisc'
discretize(object, data)
```

Arguments

object	glmdisc object
data	the data to discretize according to the provided discretization scheme

Value

A character matrix with one row per observation of data and one column per feature: the name of the level ("1", "2", ...; "NA" for a missing value) that each value was assigned to.

Author(s)

Adrien Ehrhardt.

Examples

```
# Simulation of a discretized logit model
set.seed(1)
x <- matrix(runif(300), nrow = 100, ncol = 3)
cuts <- seq(0, 1, length.out = 4)
xd <- apply(x, 2, function(col) as.numeric(cut(col, cuts)))
theta <- t(matrix(c(0, 0, 0, 2, 2, 2, -2, -2, -2), ncol = 3, nrow = 3))
log_odd <- rowSums(t(sapply(seq_along(xd[, 1]), function(row_id) {
  sapply(
    seq_along(xd[row_id, ]),
    function(element) theta[xd[row_id, element], element]
  )
})))
y <- rbinom(100, 1, 1 / (1 + exp(-log_odd)))

sem_disc <- glmdisc(x, y,
  iter = 50, m_start = 4, test = FALSE,
  validation = FALSE, criterion = "aic"
)
discretize(sem_disc, data.frame(x))
```

---

glmdisc *Model-based multivariate discretization for logistic regression*

Description

This function discretizes a training set using an SEM-Gibbs based method (see References section). It detects numerical features of the dataset and discretizes them ; values of categorical features (of type `factor`) are regrouped. This is done in a multivariate supervised way. Assessment of the correct model is done via AIC, BIC or test set error (see parameter `criterion`). Second-order interactions can be searched through the optional `interaction` parameter using a Metropolis-Hastings algorithm (see References section).

Usage

```
glmdisc(
  predictors,
  labels,
  interact = TRUE,
  validation = TRUE,
  test = TRUE,
  criterion = "gini",
  iter = 1000,
  m_start = 20,
  reg_type = "poly",
  proportions = c(0.2, 0.2),
  verbose = FALSE,
  fast = FALSE,
  patience = NULL
)
```

Arguments

`predictors`	The matrix array containing the numerical or factor attributes to discretize.
`labels`	The actual labels of the provided predictors (0/1).
`interact`	Boolean : True (default) if interaction detection is wanted (Warning: may be very memory/time-consuming).
`validation`	Boolean : True if the algorithm should use predictors to construct a validation set on which to search for the best discretization scheme using the provided criterion (default: TRUE). The coefficients of the final logistic regression are then estimated on the training and validation sets together.
`test`	Boolean : True if the algorithm should use predictors to construct a test set on which to calculate the provided criterion using the best discretization scheme (chosen thanks to the provided criterion on either the test set (if true) or the training set (otherwise)) (default: TRUE).
`criterion`	The criterion ('gini','aic','bic') to use to choose the best discretization scheme among the generated ones (default: 'gini'). Note: it is best to use 'gini' only when test is set to TRUE and 'aic' or 'bic' when it is not. When using 'aic' or 'bic' with a test set, the likelihood is returned as there is no need to penalize for generalization purposes.
`iter`	The number of iterations to do in the SEM protocol (default: 1000).

m_start	The maximum number of resulting categories for each variable wanted (default: 20).
reg_type	The model linking a continuous feature to its quantized version: multinomial logistic regression ('poly', default; fitted by a dedicated penalized Newton solver) or ordered logistic regression ('polr', polr; falls back to 'poly' when it cannot be fitted).
proportions	The proportions wanted for the test and validation sets, in this order. Not used when both test and validation are false. Only the first is used when there is only one of either test or validation that is set to TRUE. Produces an error when the sum is greater to one. Default: list(0.2,0.2) so that the training set has 0.6 of the input observations.
verbose	Print information while fitting (default: FALSE).
fast	Deprecated and ignored (sampling is now always vectorized).
patience	Optional early stopping: stop when the criterion has not improved for this many consecutive iterations (default: NULL, run all iter iterations). The SEM chain typically reaches a plateau long before the default iter; see the vignette.

Details

This function finds the most appropriate discretization scheme for logistic regression. When provided with a continuous variable X, it tries to convert it to a categorical variable Q which values uniquely correspond to intervals of the continuous variable X. When provided with a categorical variable X, it tries to find the best grouping of its values and subsequently creates categorical variable Q. The goal is to perform supervised learning with logistic regression so that you have to specify a target variable Y denoted by labels. The "discretization" process, i.e. the transformation of X to Q is done so as to achieve the best logistic regression model $p(y|e;\theta)$. It can be interpreted as a special case feature engineering algorithm. When interact = TRUE, the set of pairwise interactions is explored by a Metropolis-Hastings sampler: at each iteration one pair is proposed for addition or removal (with a probability that favors pairs for which a logistic regression on the raw features finds an interaction) and accepted according to the BIC of the logistic regression with and without it. The target of this sampler is the BIC approximation of the posterior of the set of interactions, with a Bernoulli prior given by the pairwise screening, and the proposal is corrected so that it is sampled exactly. The sampler of the thesis (BIC not halved, no proposal correction, unclamped prior) can be restored for comparison with options(glmdisc.mh_scale = 1, glmdisc.mh_hastings = FALSE, glmdisc.mh_clamp = 0). Subsequently, its outputs are: the optimal discretization scheme and the logistic regression model associated with it. We also provide the parameters that were provided to the function and the evolution of the criterion with respect to the algorithm's iterations.

Value

An S4 object of class glmdisc with the following slots:

- parameters: a list with the arguments used to fit the model (test, validation, criterion, iter, m_start, reg_type, interact), the type of each feature (types_data, "numeric" or "factor") and the levels of the factor features;
- best.disc: a list of three elements describing the best discretization scheme found: bestLogisticRegression, the glm fitted on the quantized features (and their interactions, if any); bestLinkFunction,

a list with one element per feature that describes how a raw value is assigned to a level (it is used by discretize and predict); and formulaOfBestLogisticRegression, the formula of the logistic regression;

- performance: a list with performance, the normalized Gini index of the best model on the test set (if test = TRUE, otherwise on the validation set if validation = TRUE, otherwise on the training set), and criterionEvolution, the value of the criterion at each iteration;
- disc.data: a data frame with the discretized features (levels named "1", "2", ... and "NA" for missing values) and the labels of the observations on which performance was computed; cont.data: the same observations with the raw features.

Author(s)

Adrien Ehrhardt.

References

Celeux, G., Chauveau, D., Diebolt, J. (1995), On Stochastic Versions of the EM Algorithm. [Research Report] RR-2514, INRIA. 1995. <inria-00074164>

Agresti, A. (2002) *Categorical Data*. Second edition. Wiley.

See Also

glm, polr

Examples

```
# Simulation of a discretized logit model
set.seed(1)
x <- matrix(runif(300), nrow = 100, ncol = 3)
cuts <- seq(0, 1, length.out = 4)
xd <- apply(x, 2, function(col) as.numeric(cut(col, cuts)))
theta <- t(matrix(c(0, 0, 0, 2, 2, 2, -2, -2, -2), ncol = 3, nrow = 3))
log_odd <- rowSums(t(sapply(seq_along(xd[, 1]), function(row_id) {
  sapply(
    seq_along(xd[row_id, ]),
    function(element) theta[xd[row_id, element], element]
  )
})))
y <- rbinom(100, 1, 1 / (1 + exp(-log_odd)))

sem_disc <- glmdisc(x, y,
  iter = 50, m_start = 4, test = FALSE,
  validation = FALSE, criterion = "aic"
)
print(sem_disc)
```

glmdisc-class *Class glmdisc*

Description

Class glmdisc represents a discretization scheme associated with its optimal logistic regression model.

Slots

parameters The parameters associated with the method.

best.disc The best discretization scheme found by the method given its parameters.

performance The performance obtained with the method given its parameters.

disc.data The discretized data: test set if test is TRUE; if test is FALSE and validation is TRUE, then it provides the discretized validation set. Otherwise, it provides the discretized training set.

disc.data The continuous data: test set if test is TRUE; if test is FALSE and validation is TRUE, then it provides the discretized validation set. Otherwise, it provides the discretized training set.

normalizedGini *Calculating the normalized Gini index*

Description

This function calculates the Gini index of a classification rule outputting probabilities. It is a classical metric in the context of Credit Scoring. It is equal to 2 times the AUC (Area Under ROC Curve) minus 1; tied predicted probabilities are given half credit, so the result does not depend on the order of the observations.

Usage

```
normalizedGini(actual, predicted)
```

Arguments

actual	The numeric binary vector of the actual labels observed.
predicted	The vector of the probabilities predicted by the classification rule.

Value

A single number between -1 and 1, the normalized Gini index (2 AUC - 1) of predicted for the labels actual: 1 for a perfect ranking, 0 for a ranking no better than chance, negative for an inverted ranking. It is NaN when actual contains a single class.

Author(s)

Adrien Ehrhardt

Examples

```
normalizedGini(c(1, 1, 1, 0, 0), c(0.7, 0.9, 0.5, 0.6, 0.3))
```

---

plot *Plots for the discretized / grouped data*

Description

This defines the `plot` method which will plot some useful graphs for the discretization scheme of S4 class `glmdisc`

Usage

```
## S4 method for signature 'glmdisc,missing'
plot(x, y, ...)
```

Arguments

`x`	The S4 `glmdisc` object to plot.
`y`	Not used.
`...`	Not used.

Value

No return value (`NULL`, invisibly), called for its side effects: it draws the ROC curve of the model, the risk of each level of every discretized feature, and, for numeric features, a GAM fit of the feature with its discretization. For factor features it also prints the table of the groups of levels.

Examples

```
# Simulation of a discretized logit model
set.seed(1)
x <- matrix(runif(300), nrow = 100, ncol = 3)
cuts <- seq(0, 1, length.out = 4)
xd <- apply(x, 2, function(col) as.numeric(cut(col, cuts)))
theta <- t(matrix(c(0, 0, 0, 2, 2, 2, -2, -2, -2), ncol = 3, nrow = 3))
log_odd <- rowSums(t(sapply(seq_along(xd[, 1]), function(row_id) {
  sapply(
    seq_along(xd[row_id, ]),
    function(element) theta[xd[row_id, element], element]
  )
})))
y <- rbinom(100, 1, 1 / (1 + exp(-log_odd)))
```

```
sem_disc <- glmdisc(x, y,
  iter = 50, m_start = 4, test = FALSE,
  validation = FALSE, criterion = "aic"
)
plot(sem_disc)
```

`predict` *Prediction on a raw test set of the best logistic regression model on discretized data*

Description

This defines the method "predict" which discretizes a new input dataset given a discretization scheme of S4 class `glmdisc` and returns the predicted probabilities of class 1.

Usage

```
predict(object, ...)

## S3 method for class 'glmdisc'
predict(object, predictors, ...)
```

Arguments

`object`	The S4 discretization object.
`...`	Not used.
`predictors`	The test dataframe to discretize and for which we wish to have predictions.

Details

This function discretizes a user-provided test dataset given a discretization scheme provided by an S4 "glmdisc" object. It then applies the learnt logistic regression model and outputs its prediction (see `predict.glm`).

Value

A numeric vector with the predicted probability of class 1 for each row of `predictors`.

Examples

```
# Simulation of a discretized logit model
set.seed(1)
x <- matrix(runif(300), nrow = 100, ncol = 3)
cuts <- seq(0, 1, length.out = 4)
xd <- apply(x, 2, function(col) as.numeric(cut(col, cuts)))
theta <- t(matrix(c(0, 0, 0, 2, 2, 2, -2, -2, -2), ncol = 3, nrow = 3))
log_odd <- rowSums(t(sapply(seq_along(xd[, 1]), function(row_id) {
  sapply(
```

```
    seq_along(xd[row_id, ]),
    function(element) theta[xd[row_id, element], element]
  )
}))
y <- rbinom(100, 1, 1 / (1 + exp(-log_odd)))

sem_disc <- glmdisc(x, y,
  iter = 50, m_start = 4, test = FALSE,
  validation = FALSE, criterion = "aic"
)
predict(sem_disc, data.frame(x))
```

`predictlogisticRegression` *Predicting using a logistic regression fitted with RcppNumerical::fastLR*

Description

This function returns a numeric vector containing the probability of each observation of being of class 1 given a vector of logistic regression parameters (usually estimated through RcppNumerical::fastLR).

Usage

```
predictlogisticRegression(test, parameters)
```

Arguments

`test`	A matrix containing test data
`parameters`	A vector containing the logistic regression parameters

Value

A numeric vector of length `nrow(test)`: the probability of class 1 of each row, $1/(1 + e^{-x^T \beta})$ where x is the row and β the `parameters`.

Examples

```
# intercept and one covariate: the probabilities are plogis(c(-2, 0, 2))
predictlogisticRegression(cbind(1, c(-2, 0, 2)), c(0, 1))
```

print.glmdisc *Print for the glmdisc class*

Description

Print for the glmdisc class

Usage

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

Arguments

x	S4 glmdisc object.
...	Other objects passed to print

Value

The logistic regression fitted on the discretized features (a glm object), invisibly.

Examples

```
set.seed(1)
x <- matrix(runif(300), nrow = 100, ncol = 3)
y <- rbinom(100, 1, plogis(2 * (x[, 1] > 0.5) - 1))
sem_disc <- glmdisc(x, y,
  iter = 30, m_start = 4, test = FALSE,
  validation = FALSE, criterion = "aic", interact = FALSE
)
print(sem_disc)
```

summary.glmdisc *Summary for the glmdisc class*

Description

Summary for the glmdisc class

Usage

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

Arguments

object	S4 glmdisc object.
...	Other objects passed to summary

Value

An object of class summary.glm, the summary of the logistic regression fitted on the discretized features.

Examples

```
set.seed(1)
x <- matrix(runif(300), nrow = 100, ncol = 3)
y <- rbinom(100, 1, plogis(2 * (x[, 1] > 0.5) - 1))
sem_disc <- glmdisc(x, y,
  iter = 30, m_start = 4, test = FALSE,
  validation = FALSE, criterion = "aic", interact = FALSE
)
summary(sem_disc)
```

Index

∗ **SEM Gibbs discretization**
glmdisc, 5
∗ **gini index**
normalizedGini, 9
∗ **test discretization plot**
plot, 10
∗ **test discretization predict prediction**
predict, 11

cutpoints, *3*, 3
cutpoints,glmdisc-method *(*cutpoints*)*, 3

discretize, *3*, 4, *8*
discretize,glmdisc-method *(*discretize*)*, 4

glm, *7, 8*, *13*
glmdisc, *2–4*, 5, *7*, *10, 11*
glmdisc-class, 9
glmdisc-package, 2

normalizedGini, *3*, 9

plot, *10*, 10
plot,glmdisc,missing-method *(*plot*)*, 10
polr, *7, 8*
predict, *3*, *8*, 11
predict,glmdisc-method *(*predict*)*, 11
predict.glm, *4*, *11*
predict.glmdisc *(*predict*)*, 11
predictlogisticRegression, 12
print, *3*
print.glmdisc, 13

show, *3*
summary, *3*
summary.glmdisc, 13