Package ‘glmdisc’

March 22, 2020

Type Package

Title Discretization and Grouping for Logistic Regression

Version 0.5

Date 2020-02-20

Maintainer Adrien Ehrhardt <adrien.ehrhardt@centraliens-lille.org>

Description A Stochastic-Expectation-Maximization (SEM) algorithm (Celeux et al. (1995) <https://hal.inria.fr/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) <arXiv:1903.08920>). Continuous features are discretized and categorical features' values are grouped to produce a better logistic regression model. Pairwise interactions between quantized 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

LazyData true

Imports caret (>= 6.0-82), gam, nnet, RcppNumerical, methods, MASS, graphics, Rcpp (>= 0.12.13)

LinkingTo Rcpp, RcppEigen, RcppNumerical

URL https://adimajo.github.io

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

RoxygenNote 7.0.2

Suggests knitr, rmarkdown, testthat, covr

VignetteBuilder knitr

Collate 'RcppExports.R' 'allClasses.R' 'cut.dataset.R' 'discretize.link.R' 'generic_cutpoints.R' 'generic_discretize.R' 'glmdisc.R' 'method_cutpoints.R' 'method_discretize.R' 'method_plot.R' 'method_predict.R' 'methods_disc.R' 'normalizedGini.R' 'semDiscretization.R'

NeedsCompilation yes

Author Adrien Ehrhardt [aut, cre],
Vincent Vandewalle [aut],
Christophe Biernacki [ctb],
Philippe Heinrich [ctb]

Repository CRAN

Date/Publication 2020-03-22 11:30:02 UTC

R topics documented:

`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.

---

cutpoints *Obtaining the cutpoints and / or regroupments 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.

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

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)
)
```

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).

`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'). Nota Bene: 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 protocole (default: 1000).

`m_start` The maximum number of resulting categories for each variable wanted (default: 20).

`reg_type` The model to use between discretized and continuous features (currently, only multinomial logistic regression ('poly') and ordered logistic regression ('polr') are supported ; default: 'poly'). WARNING: 'poly' requires the `mnlogit` package, 'polr' requires the `MASS` package.

`proportions` The list of the proportions wanted for test and validation set. 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.

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 regroupement 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. 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.

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`, `multinom`, `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.

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.

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

```
plot.glmdisc(x)

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

Arguments

x	The S4 glmdisc object to plot.

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 "discretize" which will discretize a new input dataset given a discretization scheme of S4 class `glmdisc`

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

Usage

```
predict(object, ...)

predict.glmdisc(object, predictors)

## S4 method for signature 'glmdisc'
predict(object, predictors)
```

Arguments

`object`	The S4 discretization object.
`...`	Essai
`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`).

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`).

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 RCpp::fast_LR.

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 RCpp::fast_LR).

Usage

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

Arguments

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

Index