Package 'CatReg'

October 12, 2022

Type Package

Title Solution Paths for Linear and Logistic Regression Models with Categorical Predictors, with SCOPE Penalty

Version 2.0.3

Date 2021-06-08

Description Computes solutions for linear and logistic regression models with potentially high-dimensional categorical predictors. This is done by applying a nonconvex penalty (SCOPE) and computing solutions in an efficient path-wise fashion. The scaling of the solution paths is selected automatically. Includes functionality for selecting tuning parameter lambda by k-fold cross-validation and early termination based on information criteria. Solutions are computed by cyclical block-coordinate descent, iterating an innovative dynamic programming algorithm to compute exact solutions for each block.

License GPL (>= 2)

Imports Rcpp (>= 1.0.1),Rdpack

LinkingTo Rcpp

NeedsCompilation yes

RdMacros Rdpack

RoxygenNote 7.1.1

Author Benjamin Stokell [aut],
Daniel Grose [ctb, cre],
Rajen Shah [ctb]

Maintainer Daniel Grose <dan.grose@lancaster.ac.uk>

Repository CRAN

Date/Publication 2021-06-14 13:50:32 UTC

R topics documented:

`CorrelatedDesignMatrix` *Create a design matrix of categorical variables with correlated columns.*

Description

Function for use in simulations. Created design matrix of categorical variables with correlated columns

Usage

```
CorrelatedDesignMatrix(n, cov_mat, c)
```

Arguments

`n`	Number of observations
`cov_mat`	p x p covariance matrix. Controls correlations of pairs of marginally U[0,1] random variables that are subsequently binned to assign categories for each variable
`c`	Number of categories within each variable

Value

A data frame of categorical (factor) variables.

Examples

```
# Generate matrix of marginal U[0,1] variables, 0.5 pairwise correlation, that are
# discretised into factor variables
cov_mat = 0.5 * diag(10) + 0.5 * matrix(1, 10, 10)
x = CorrelatedDesignMatrix(100, cov_mat, 8)
```

`predict.scope` *Computes SCOPE predictions*

Description

Computes SCOPE predictions on new data.

Usage

```
## S3 method for class 'scope'
predict(object, newdata, include_intercept = TRUE, ...)
```

Arguments

`object`	SCOPE model as outputted by scope. Must have simply.the.best = TRUE
`newdata`	New covariates on which to make predictions. Must be of the same form as the model was trained on
`include_intercept`	If TRUE, a column of 1s will be added to the (continuous) design matrix. Must match format of training data.
`...`	Additional arguments to pass to other `predict` methods

Value

Returns n-length vector of predictions

See Also

`scope`

`predict.scope.logistic` *Computes SCOPE logistic predictions*

Description

Computes SCOPE logistic predictions on new data

Usage

```
## S3 method for class 'scope.logistic'
predict(object, newdata, probs = TRUE, include_intercept = TRUE, ...)
```

Arguments

`object`	SCOPE model as outputted by scope.logistic. Must have simply.the.best = TRUE
`newdata`	New covariates on which to make predictions. Must be of the same form as the model was trained on
`probs`	If TRUE returns probabilities, if FALSE returns binary predictions
`include_intercept`	If TRUE, a column of 1s will be added to the (continuous) design matrix. Must match format of training data.
`...`	Additional arguments to pass to other `predict` methods

Value

Returns n-length vector of predictions

See Also

scope.logistic

---

scope *Compute solution for SCOPE linear models.*

Description

Computes solution for SCOPE linear models. Performs K-fold cross-validation for regularisation parameter lambda and can incorporate both linear and categorical (including logical) variables.

Usage

```
scope(
  x,
  y,
  gamma = 8,
  lambda = NULL,
  nlambda = 100,
  lambda_min_ratio = 0.01,
  nfolds = 5,
  include_intercept = TRUE,
  return_full_beta = FALSE,
  max_iter = 1000,
  early_stopping = ifelse(pshrink > 1, TRUE, FALSE),
  early_stopping_rounds = 20,
  early_stopping_criterion = "AIC",
  noise_variance = NULL,
  terminate_eps = 1e-07,
  silent = TRUE,
  only_cross_validate = FALSE,
  grid_safe = 10,
  block_order = NULL,
  fold_assignment = NULL
)
```

Arguments

x	Data frame of covariates: Can include a mix of continuous and categorical variables (no scaling of continuous covariates is performed within the program). By default an intercept will be added to the linear part; see include_intercept
y	Response vector of length n
gamma	Concavity parameter in MCP; see Zhang (2010)
lambda	If NULL default sequence will be generated. Matrix of values (p_categorical times nlambda) of penalty parameter lambda. Must be non-negative and each row decreasing. Note that if lambda = 0 then no shrinkage will occur.

`nlambda` Length of default sequence of lambda values generated if lambda = NULL

`lambda_min_ratio`
Ratio of largest to smallest value on default sequence of lambda values

`nfolds` Number of folds in cross-validation. If nfolds = 1, no cross-validation is performed

`include_intercept`
If TRUE, a column of 1s will be added to the (continuous) design matrix

`return_full_beta`
If TRUE with cross-validation, the entire solution path will be returned instead of just the optimal point

`max_iter` Maximum number of iterations at each point on the lambda path

`early_stopping` Early stopping based on information criterion. By default is TRUE if there are more than 1 categorical variables

`early_stopping_rounds`
Number of iterations that information criterion must have not decreased for to terminate

`early_stopping_criterion`
If "AIC", Akaike Information Criterion is used for early stopping. Otherwise if a positive number is given, modified Bayes Information Criterion is used with this integer as the parameter (Wang et al., 2009)

`noise_variance` If noise variance is known, this will be used for scaling the default values of lambda. Otherwise this will be scaled automatically

`terminate_eps` Epsilon for convergence criterion, is multiplied by null deviance to get terminate criterion for objective value

`silent` If FALSE then progress updates will be printed as solutions are computed. Useful for tuning and diagnosing convergence issues.

`only_cross_validate`
If TRUE then cross-validation scores for each value of lambda will be returned, but not the estimates themselves

`grid_safe` As the automatically generated sequence of lambda values is adjusted during the first fold but fixed thereafter. For subsequent folds, this sets computation to begin at a larger value of lambda to ensure that the first solution along the path is zero so as to maintain the advantages of the pathwise approach. This specifies how many larger values of lambda should be used

`block_order` By default the order in block coordinate descent is randomly sampled. Alternatively a permutation vector can be included here

`fold_assignment`
By default the assignments for cross-validation are randomly sampled automatically. Alternatively assignments can be included here

Value

A list of objects. Some may not be returned depending on value of arguments K, simply.cross.validated, return.full.beta.

- lambda - A matrix of the values of lambda used to compute the solution path. Columns correspond to different points on the path, rows correspond to the categorical variables. Lambda is scaled depending on the number of categories present in the data.
- cverrors - Provided nfolds > 1 then the cross-validation error for each point on the grid will be returned
- beta.full - Contains full solution path. If nfolds > 1 then will only be returned if simply.cross.validated = FALSE and return.full.beta = TRUE. First object [[1]] is coefficients of continuous variables, [[2]] is a list of coefficients for categorical variables
- beta.best - Contains solution at CV-optimal point. Requires nfolds > 1 to be returned. This must not be NULL in order to use predict.scope. First object [[1]] is coefficients of continuous variables, [[2]] is a list of coefficients for categorical variables
- fold.assign - Contains fold assignments for cross-validation

References

Zhang C (2010). "Nearly unbiased variable selection under minimax concave penalty." *The Annals of Statistics*, **38**(2). ISSN 0090-5364, doi: 10.1214/09AOS729.

Wang H, Li B, Leng C (2009). "Shrinkage tuning parameter selection with a diverging number of parameters." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **71**(3), 671–683. doi: 10.1111/j.14679868.2008.00693.x, https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9868.2008.00693.x.

Examples

```
set.seed(1)
x = UniformDesignMatrix(100, 5, 8)
y = (as.integer(x[ , 1 ]) < 5) + (as.integer(x[ , 2 ]) < 5)  + rnorm(100)
scope_mod = scope(x, y)
x_new = UniformDesignMatrix (10, 5, 8)
predict(scope_mod, x_new)
```

---

scope.logistic *Computes solution for SCOPE logistic models*

Description

Computes solution for SCOPE logistic models, computing along a path and iterating local quadratic approximations at each point. Performs K-fold cross-validation for regularisation parameter lambda and can incorporate both linear and categorical (including logical) variables. This function uses a Proximal Newton algorithm and is not guaranteed to converge. It is recommended for developer use only.

Usage

```
scope.logistic(
  x,
  y,
  gamma = 8,
  lambda = NULL,
  nlambda = 100,
  lambda_min_ratio = 0.01,
  nfolds = 5,
  include_intercept = TRUE,
  return_full_beta = FALSE,
  max_iter = 1000,
  max_out_iter = 1000,
  early_stopping = ifelse(pshrink > 1, TRUE, FALSE),
  early_stopping_rounds = 20,
  early_stopping_criterion = "AIC",
  noise_variance = NULL,
  terminate_eps = 1e-07,
  silent = TRUE,
  only_cross_validate = FALSE,
  grid_safe = 10,
  block_order = NULL,
  fold_assignment = NULL,
  zero_penalty = FALSE
)
```

Arguments

x	Data frame of covariates: Can include a mix of continuous and categorical variables (no scaling of continuous covariates is performed within the program). By default an intercept will be added to the linear part; see include_intercept
y	Response vector of length n
gamma	Concavity parameter in MCP; see Zhang (2010)
lambda	If NULL default sequence will be generated. Matrix of values (p_categorical times nlambda) of penalty parameter lambda. Must be non-negative and each row decreasing. Note that if lambda = 0 then no shrinkage will occur.
nlambda	Length of default sequence of lambda values generated if lambda = NULL
lambda_min_ratio	Ratio of largest to smallest value on default sequence of lambda values
nfolds	Number of folds in cross-validation. If nfolds = 1, no cross-validation is performed
include_intercept	If TRUE, a column of 1s will be added to the (continuous) design matrix
return_full_beta	If TRUE with cross-validation, the entire solution path will be returned instead of just the optimal point

`max_iter` Maximum number of iterations at each local quadratic approximation

`max_out_iter` Maximum number of local quadratic approximations at each value of lambda

`early_stopping` Early stopping based on information criterion. By default is TRUE if there are more than 1 categorical variables

`early_stopping_rounds`
Number of iterations that information criterion must have not decreased for to terminate

`early_stopping_criterion`
If "AIC", Akaike Information Criterion is used for early stopping. Otherwise if a positive number is given, modified Bayes Information Criterion is used with this integer as the parameter (Wang et al., 2009)

`noise_variance` If noise variance is known, this will be used for scaling the default values of lambda. Otherwise this will be scaled automatically

`terminate_eps` Epsilon for convergence criterion, is multiplied by null deviance to get terminate criterion for objective value

`silent` If FALSE then progress updates will be printed as solutions are computed. Useful for tuning and diagnosing convergence issues.

`only_cross_validate`
If TRUE then cross-validation scores for each value of lambda will be returned, but not the estimates themselves

`grid_safe` As the automatically generated sequence of lambda values is adjusted during the first fold but fixed thereafter. For subsequent folds, this sets computation to begin at a larger value of lambda to ensure that the first solution along the path is zero so as to maintain the advantages of the pathwise approach. This specifies how many larger values of lambda should be used

`block_order` By default the order in block coordinate descent is randomly sampled. Alternatively a permutation vector can be included here

`fold_assignment`
By default the assignments for cross-validation are randomly sampled automatically. Alternatively assignments can be included here

`zero_penalty` Fits unpenalised logistic regression model. Used for testing purposes only.

Value

A list of objects. Some may not be returned depending on value of arguments K, simply.cross.validated, return.full.beta.

- lambda - A matrix of the values of lambda used to compute the solution path. Columns correspond to different points on the path, rows correspond to the categorical variables. Lambda is scaled depending on the number of categories present in the data.
- cverrors - Provided nfolds > 1 then the cross-validation error for each point on the grid will be returned
- beta.full - Contains full solution path. If nfolds > 1 then will only be returned if simply.cross.validated = FALSE and return.full.beta = TRUE. First object [[1]] is coefficients of continuous variables, [[2]] is a list of coefficients for categorical variables

- beta.best - Contains solution at CV-optimal point. Requires nfolds > 1 to be returned. This must not be NULL in order to use predict.scope. First object [[1]] is coefficients of continuous variables, [[2]] is a list of coefficients for categorical variables
- fold.assign - Contains fold assignments for cross-validation

References

Zhang C (2010). "Nearly unbiased variable selection under minimax concave penalty." *The Annals of Statistics*, **38**(2). ISSN 0090-5364, doi: 10.1214/09AOS729.

Wang H, Li B, Leng C (2009). "Shrinkage tuning parameter selection with a diverging number of parameters." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **71**(3), 671–683. doi: 10.1111/j.14679868.2008.00693.x, https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9868.2008.00693.x.

Examples

```
## Not run:
x = UniformDesignMatrix(200, 5, 5)
y = (as.integer(x[ , 1 ]) < 3) + rnorm(200)
y = as.integer(y > 0.8)
scope_mod = scope.logistic(x, y)
x_new = UniformDesignMatrix(10, 5, 5)
predict(scope_mod, x_new)

## End(Not run)
```

`UniformDesignMatrix` *Create a design matrix of categorical variables.*

Description

Function for use in simulations, creating design matrix of categorical variables where each column is uniformly randomly distributed and independent.

Usage

```
UniformDesignMatrix(n, p, c)
```

Arguments

n	Number of observations
p	Number of variables
c	Number of categories within each variable

Value

A data frame of categorical (factor) variables.

Examples

```
x = UniformDesignMatrix(100, 10, 8)
```

Index