Package ‘seqest’

October 14, 2022

Type Package

Title Sequential Method for Classification and Generalized Estimating Equations Problem

Version 1.0.1

Maintainer Xiaoba Pan <july666@mail.ustc.edu.cn>

Description Sequential method to solve the the binary classification problem by Wang (2019) <arXiv:arXiv:1901.10079>, multi-class classification problem by Li (2020) <doi:10.1016/j.csda.2020.106911> and the highly stratified multiple-response problem by Chen (2019) <doi:10.1111/biom.13160>.

NeedsCompilation yes

License GPL (>= 2)

Encoding UTF-8

LazyData true

Imports Rcpp (>= 1.0.2), geepack, mvtnorm, nnet, VGAM, MASS, foreach, stats

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 6.1.1

Repository CRAN

Date 2020-06-10 07:55:10 UTC

Author Xiaoba Pan [aut, cre]

Date/Publication 2020-06-17 13:50:02 UTC

R topics documented:

`ase_seq_logit` *variable selection and stopping criterion*

Description

`ase_seq_logit` determine the effective variables and whether to stop selecting samples

Usage

```
ase_seq_logit(X, Y, intercept = FALSE, criterion = "BIC", d = 0.5,
  alpha = 0.95, gamma = 1, eta = 0.75, upper = 2, lower = 0.1,
  divid.num = 10)
```

Arguments

`X`	A dataframe that each row is a sample,each column represents an independent variable.
`Y`	Numeric vector consists of 0 or 1. The length of Y must be the same as the X.
`intercept`	A logical value indicating whether add intercept to model. The default value is FALSE.
`criterion`	For the "chosfun" methods, a character string that determines the model selection criterion to be used, matching one of 'BIC' or 'AIC. The default value is 'BIC'.

d	A numeric number specifying the length of the fixed size confidence set for our model. Note that the smaller the d, the larger the sample size and the longer the time costs. The default value is 0.5.
alpha	A numeric number used in the chi-square distribution. The default value is 0.95.
gamma	A numeric number to determine the effective variables with eta. The default value is 1.
eta	A numeric number to determine the effective variables with gamma. The default value is 0.75.
upper	A numeric number to choose the right epsilon with params lower and divide.num. The value of upper should be larger than lower. The default value is 2.
lower	A numeric number to choose the right epsilon with params upper and divide.num. The default value is 0.1.
divid.num	A numeric number to choose the right epsilon with params upper and lower. Note that it should be a integer. The default value is 10.

Details

ase_seq_logit estimates the logistic regression coefficient and determines the effecrive variables and decides whether to stop selecting samples based on the current sample and its corresponding label. The parameters 'upper', 'lower' and 'divid.num' is used to get different epsilons. If different epsilons get the same value, we choose the smallest epsilon.

Value

a list containing the following components

N	current sample size
is_stopped	the label of sequential stop or not. When the value of is_stopped is 1, it means the iteration stops
betahat	the estimated coefficients based on current X and Y. Note that some coefficient will be zero. These are the non-effectiva variables should be ignored.
cov	the covariance matrix between variables
phat	the number of effective varriables.
ak	1-alpha quantile of chisquare distribution with degree of freedom phat
lamdmax	the maximum eigenvalue based on the covariance of data

---

`A_optimal_cat` *Get the most informative subjects from unlabeled dataset for the categorical case*

Description

Get the most informative subjects from unlabeled dataset under the categorical case.

Usage

```
A_optimal_cat(X, beta, W, unlabeledIDs)
```

Arguments

`X`	A matrix containing all the samples except their labels including the labeled samples and the unlabeled samples.
`beta`	A matrix contains the estimated coefficient. Note that the beta is a n * k matrix which n is the number of the explanatory variables and k+1 is the number of categories
`W`	A matrix denotes the inverse information matrix of the coefficient beta.
`unlabeledIDs`	A numeric vector for the unique identification of the unlabeled. dataset.

Details

A_optimal_cat uses the A optimality criterion from the experimental design to choose the most informative subjects under the the categorical case. We have obtained the variance-covariance matrix based on the current labeled samples which indicates how much information there is. Then we should repeatly calculate the information matrix after we choose a sample from the unlabeled dataset. Once we finish the iteration, we pick the sample which has the most information.

Value

a index of the most informative subjects from unlabeled dataset for the categorical case

---

`A_optimal_ord` *Get the most informative subjects from unlabeled dataset for the ordinal case*

Description

Get the most informative subjects from unlabeled dataset under the ordinal case

Usage

```
A_optimal_ord(X, beta, W, unlabeledIDs)
```

Arguments

`X`	A matrix containing all the samples except their labels including the labeled samples and the unlabeled samples.
`beta`	A matrix contains the estimated coefficient. Note that the beta is a n * k matrix which n is the number of the explanatory variables and k+1 is the number of categories
`W`	A matrix denotes the inverse information matrix of the coefficient beta.
`unlabeledIDs`	A numeric vector for the unique identification of the unlabeled. dataset.

Details

A_optimal_ord uses the A optimality criterion from the experimental design to choose the most informative subjects under the the ordinal case. We have obtained the variance-covariance matrix based on the current labeled samples which indicates how much information there is. Then we should repeatly calculate the information matrix after we choose a sample from the unlabeled dataset. Once we finish the iteration, we pick the sample which has the most information.

Value

a index of the most informative subjects from unlabeled dataset for the ordinal case

---

`D_optimal` *Get the most informative subjects for the clustered data*

Description

Get the most informative subjects for the highly stratified response data by the D-optimility.

Usage

```
D_optimal(X, id, beta, nonZeroSet, M, rho, linkv, corstrv)
```

Arguments

`X`	A data frame contains all the random samples which we will choose subject from.
`id`	The id for each subject in the X
`beta`	The paramters that we estimate under the current samples
`nonZeroSet`	The set of the index of the non zero coefficient
`M`	A numeric matrix calculated by the `getMH` function
`rho`	A numeric number indicating the estimate of correlation coefficient
`linkv`	A specification for the model link function.
`corstrv`	A character string specifying the correlation structure. The following are permitted: "independence", "exchangeable" and "ar1".

Details

D_optimal uses the D-optimality criterion from the experimental design to choose the most informative subjects for the highly stratified response data.

Value

a index of the most informative subject

`evaluateGEEModel` *The adaptive shrinkage estimate for generalized estimating equations*

Description

`evaluateGEEModel` is used to get a generalized estimating equation of the data by the adaptive shrinkage estimate method.

Usage

```
evaluateGEEModel(family, corstr, y, x, clusterID, criterion = "QIC",
  theta = 0.75, gamma = 1, leastVar = 3, mostVar = ncol(x))
```

Arguments

`family`	A description of the error distribution and link function to be used in the model. See family for details of `family` functions.
`corstr`	A character string specifying the correlation structure. The following are permitted: "independence", "exchangeable" and "ar1".
`y`	The response data.
`x`	A data frame contains the covariate vectors.
`clusterID`	The id for each subject in the initial samples. Note that the subjects in the same cluster will have identical id.
`criterion`	The model selection criteria, one of the 'PWD' or 'QIC'.
`theta`	The parameters of the adaptive shrinkage estimate.
`gamma`	The parameters of the adaptive shrinkage estimate.
`leastVar`	The minimum number of variables.
`mostVar`	The maximum number of variables.

Details

evaluateGEEModel fits the current data by generalized estimating equations(GEE) according to the value of the family argument and the corstr argument. We should notice that this is not the ordinary generalized estimating equations. It can determine the variables that have an impact on the response which called effective variables. We use model selection criteria like the QIC criterion to choose the optimal value.

Value

a list containing the following components

`rho`	the correlation coefficient of the clusters
`beta`	parameters that we estimate under the current samples
`sandwich`	the sandwich information matrix for covariance
`nonZeroIdx`	the index of the non zero coefficients
`call`	a list containing several matrices including the sandwich matrix

genBin *Generate the correlated binary response data for discrete case*

Description

genBin generate the data used for discrete case

Usage

```
genBin(corstr, mu = NULL, size = NULL)
```

Arguments

corstr	A character string specifying the correlation structure for the clusters. Allowed structures are: "independence", "exchangeable" and "ar1".
mu	A numeric parameter denotes the value of the link function
size	A numeric number indicating the size of the matrix.

Details

genBin returns the correlated binary response according to the value of the corstr argument.

Value

a function to get the correlated binary response data

genCorMat *Generate the correlation matrix for the clusteded data*

Description

genCorMat generate the data with specified correlation matrix.

Usage

```
genCorMat(corstr, rho, size)
```

Arguments

corstr	A character string specifying the correlation structure for the clusters. Allowed structures are: "independence", "exchangeable" and "ar1".
rho	A numeric parameter in correlation structure for the autocorrelation coefficient.
size	A numeric number indicating the size of the matrix.

Details

genCorMat returns the corresponding correlation matrix according to the value of the corstr argument.

Value

a matrix which represents the diffetent correlation matrix.

`gen_bin_data` *generate the data used for the model experiment*

Description

`gen_bin_data` generate the data used for the model experiment

Usage

```
gen_bin_data(beta, N, nclass, seed)
```

Arguments

`beta`	A numeric vector that represents the true coefficients that used to generate the synthesized data.
`N`	A numeric number specifying the number of the synthesized data. It should be an integer.
`nclass`	A numeric number used to specify how many clusters the original data would be transformed into. It should be an integer.
`seed`	Set random number seed.

Details

The function gen_bin_data generates N points. That is,the first column of the design matrix is 1 and the second column has a normal distribution with a mean of 1 and a variance of 1 and the rest columns with a mean of 0 and a variance of 1. Next, they are clustered into classes to decrease the computation cost. You should specify the number of classes. In the function, it's the parameter nclass.

Value

a list of seven elements:

`data.clust`	list with clustering results. Samples in the same list element are closer with each other
`X`	the samples with the smallest variance from each cluster. Note that the length of X is the same as the number of data.clust
`y`	the target value of 0 or 1 corresponding to X

References

Wang Z, Kwon Y, Chang YcI (2019). Active learning for binary classification with variable selection. arXiv preprint arXiv:1901.10079.

See Also

`gen_multi_data` for categorical and ordinal case

`gen_GEE_data` for generalized estimating equations case.

Examples

```
# For an example, see example(seq_bin_model)
```

---

gen_GEE_data *Generate the datasets with clusters*

Description

`gen_GEE_data` generates the clustered data used for the generalized estimating equations with sequential method.

Usage

```
gen_GEE_data(numClusters, clusterSize, clusterRho, clusterCorstr, beta,
  family, intercept = TRUE, xCorstr = "ar1", xCorRho = 0.5,
  xVariance = 0.2)
```

Arguments

`numClusters`	A numeric number represents the number of clusters we will generated. Note that each cluster has several similar subjects. It should be a integer.
`clusterSize`	A numeric number specifying the number of subjects in each cluster. The subject in the same cluster is highly correlated to each other which can be regarded as the longitudinal data.
`clusterRho`	A numeric parameter in correlation structure for the clusters. It will be ignored when responseCorstr is independence.
`clusterCorstr`	A character string specifying the correlation structure for the clusters. Allowed structures are: "independence", "exchangeable" and "ar1".
`beta`	A nummeric vector denotes the true parameter in GEE model.
`family`	The type of response data, matching one of 'gaussian()' or 'binomial()'. The 'gaussian()' corresponds to the continuous case and 'binomial' corresponds to the discrete case.
`intercept`	A logical value indicating whether to add intercept term. The default value is TRUE.

`xCorstr`	A character string specifying the correlation structure for the covariate. The default value is 'ar1'.
`xCorRho`	A numeric parameter indicating the correlation coefficient in covariables. It does something similar to what the argument clusterRho does. The default value is 0.5.
`xVariance`	A numeric number specifying the marginal variance in the correlation matrix in one clusters. The default value is 0.2.

Details

The gen_GEE_data function is used to generate data. We can get data from two different distributions, corresponding to continuous and discrete cases. In the continuous case, the covariates vector x is created from a multivariate normal distribution with mean 0 and an AR(1) correlation matrix with autocorrelation coefficient and marginal variance. The value of autocorrelation coefficient and marginal variance are two arguments which we need specified. Then, the response y is generated by the equation: y = wx + e where the random error vector e follows a normal distribution with mean 0 and three different covariance structures with corresponding dimensional numbers. These three covariance matrices are the identity matrix, the exchangeable, and the AR(1) autoregressive correlation structure. In the discrete case, we use a logistic model. The covariates vectors x is the same as the continuous case. The binary response vector for each cluster has an AR(1) correlation structure with correlation coefficient alpha, and the marginal expectation u satisfies the following equation: logit(u) = wx

Value

a list containing the following components

`x`	the covariate matrices. Note that the number of rows is numClusters * clusterSize and the number of columns is the length of beta + 1 if intercept is TRUE.
`y`	the response data which has the same number of rows to x
`clusterID`	the id for each sample. Note that the subjects in the same cluster will have identical id.

References

Chen, Z., Wang, Z., & Chang, Y. I. (2019). Sequential adaptive variables and subject selection for GEE methods. *Biometrics*. doi:10.1111/biom.13160

See Also

`gen_multi_data` for categorical and ordinal case

`gen_bin_data` for binary classification case.

Examples

```
initialSampleSize <-  75
clusterSize <-  5
responseCorstr <-  "ar1"
responseCorRho <-  0.3
```

```
response <-  gaussian()
beta0 <-  c(1, -1.1, 1.5, -2, rep(0, 50))
xVariance <-  0.2
xCorRho <-  0.5
xCorstr <-  "ar1"
data <- gen_GEE_data(numClusters = initialSampleSize,
                     clusterSize = clusterSize,
                     clusterCorstr = responseCorstr,
                     clusterRho = responseCorRho,
                     beta = beta0,
                     family = response,
                     intercept = TRUE,
                     xVariance = xVariance,
                     xCorstr = xCorstr,
                     xCorRho = xCorRho)
```

`gen_multi_data` *Generate the training data and testing data for the categorical and ordinal case.*

Description

`gen_multi_data` generate the data used for multiple-class classification problems.

Usage

```
gen_multi_data(beta0, N, type, test_ratio)
```

Arguments

`beta0`	A numeric matrix that represent the true coefficient that used to generate the synthesized data.
`N`	A numeric number specifying the number of the synthesized data. It should be a integer. Note that the value shouldn't be too small. We recommend that the value be 10000.
`type`	A character string that determines which type of data will be generated, matching one of 'ord' or 'cat'.
`test_ratio`	A numeric number specifying proportion of test sets in all data. It should be a number between 0 and 1. Note that the value of the test_ratio should not be too large, it is best if this value is equal to 0.2-0.3.

Details

gen_multi_data creates training dataset and testing datasets. The beta0 is a p * k matrix which p is the length of true coefficient and (k + 1) represents the number of categories. The value of 'type' can be 'ord' or 'cat' . If it equals to 'ord', it means the data has an ordinal relation among classes ,which is common in applications (e.g., the label indicates the severity of a disease or product preference). If it is 'cat', it represents there is no such ordinal relations among classes. In addition, the response variable y are then generated from a multinomial distribution with the explanatory variables x generated from a multivariate normal distribution with mean vector equal to 0 and the identity covariance matrix.

Value

a list containing the following components

`train_id`	The id of the training samples
`train`	the training datasets. Note that the id of the data in the train dataset is the same as the train_id
`test`	the testing datasets

References

Li, J., Chen, Z., Wang, Z., & Chang, Y. I. (2020). Active learning in multiple-class classification problems via individualized binary models. *Computational Statistics & Data Analysis*, 145, 106911. doi:10.1016/j.csda.2020.106911

See Also

`gen_bin_data` for binary classification case

`gen_GEE_data` for generalized estimating equations case.

Examples

```
# For an example, see example(seq_ord_model)
```

`getMH` *Get the matrices M and H for the clustered data for the GEE case*

Description

Get the matrices M and H to approximate the true covariance matrix of the GEE case

Usage

```
getMH(y, X, id, beta, rho, linkv, corstrv)
```

Arguments

`y`	A matrix containing current response variable
`X`	A data frame containing the covariate for the current samples
`id`	The id for each subject in the X
`beta`	The paramters that we estimate when we use the current samples
`rho`	A numeric number indicating the estimate of correlation coefficient
`linkv`	A specification for the model link function.
`corstrv`	A character string specifying the correlation structure. The following are permitted: "independence", "exchangeable" and "ar1".

Details

getMH uses the current samples to obtain the covariance matrix.

Value

a list contains several components

`sandwich`	the sandwich information matrix for covariance
`M`	the matrix for calculating the sandwich information matrix for covariance
`H`	the matrix for calculating the sandwich information matrix for covariance

---

`getWH` *Get the matrices W and H for the categorical case*

Description

Get the matrices W and H using the Rcpp package for the categorical case

Usage

```
getWH(data, beta)
```

Arguments

`data`	A matrix containing the training samples which we will use in the categorical case.
`beta`	A matrix contains the estimated coefficient. Note that the beta_mat is a n * k matrix which n is the number of the explanatory variables and k+1 is the number of categories

Details

getWH uses the current training data and the estimated coefficient under the categorical case to obtain the matrices W and H to further get the variance-covariance matrix and minimum eigenvalue. The variance-covariance matrix and minimum eigenvalue will be used in the process of selecting the new sampling and determining whether to stop the iteration. Note that using the Rcpp package can significantly reduce the time of operation and get conclusions faster.

Value

a list contains several components including the variance-covariance matrix, minimum eigenvalue, W and H.

getWH_ord *Get the matrices W and H for the ordinal case*

Description

Get the matrices W and H using the Rcpp package for the ordinal case

Usage

```
getWH_ord(data, beta)
```

Arguments

`data`	A matrix containing the training samples which we will use in the ordinal case.
`beta`	A matrix contains the estimated coefficient. Note that the beta_mat is a n * k matrix which n is the number of the explanatory variables and k+1 is the number of categories

Details

getWH_ord uses the current training data and the estimated coefficient under the ordinal case to obtain the matrices W and H to further get the variance-covariance matrix and minimum eigenvalue. Note that using the Rcpp package can significantly reduce the time of operation and get conclusions faster.

Value

a list contains several components including the variance-covariance matrix, minimum eigenvalue, W and H.

init_multi_data *Generate the labeled and unlabeled datasets*

Description

`init_multi_data` creates the labeled and unlabeled datasets for the categorical and ordinal case.

Usage

```
init_multi_data(train_id, train, init_N, type)
```

Arguments

`train_id` A numeric vector denotes the id of the all training samples. Each sample corresponds to a unique identification from 1 to the length of all the samples.

`train` A numeric matrix denote the training datasets. The length of the train's row is the number of the training samples and the first column represents the labels and the rest columns are the explanatory variables. Note that the id of the sample in the train dataset is the same as the train_id.

`init_N` A numeric value that determine the number of the initial labeled samples. Note that it shouldn't be too large or too small.

`type` A character string that determines which type of data will be generated, matching one of 'ord' or 'cat'.

Details

init_multi_data generates the initial labeled dataset and the unlabeled datasets which we will select a most informative sample from the unlabeled datasets into the labeled dataset. The number of samples in the initial labeled datasets is specified the init_N argument. The value of 'type' should be'ord' or 'cat'. If it equals to 'ord', the element of the splitted will be composed of samples from Classes K and Classes K+1. Otherwise, the element of the splitted will be composed of samples from Classes 0 and Classes K.

Value

a list containing the following components

`splitted` a list containing the datasets which we will use

`train` the initial labeled datasets. The number of the datasets is specified by the init_N

`newY` the value of the labels from 0 to K which denotes the number of categories

`labeled_ids` the unique id of the initial labeled dataset

`unlabeled_ids` the unique id of the unlabeled dataset

`data` the all training samples which is composed of the samples corresponding to labeled_ids and samples corresponding to unlabeled_ids

Examples

```
## For an example, see example(seq_ord_model)
```

`is_stop_ASE` *Determining whether to stop choosing sample*

Description

`is_stop_ASE` determines whether to stop choosing sample based on the current estimator

Usage

```
is_stop_ASE(sandwich, d, nonZeroIdx, verbose = FALSE)
```

Arguments

`sandwich`	A numeric matrix that represent the sandwich information matrix for covariance
`d`	A numeric number specifying the length of the fixed size confidence set that we specify
`nonZeroIdx`	A numeric number specifying the index of the non zero coefficient
`verbose`	A A logical value to determine whether to get the full information about the iteration situation

Details

is_stop_ASE determines if the iteration stop condition is met based on the current estimator

Value

a list of these elements:

`stop`	a logical value. If TRUE, it means we have choosen enough samples.
`eigen`	the maximum eighevalue covariance matrix

`logit_model` *the individualized binary logistic regression for categorical response data.*

Description

`logit_model` fit the categorical data by the individualized binary logistic regression

Usage

```
logit_model(splitted, newY)
```

Arguments

`splitted`	A list containing the datasets which we will use in the categorical case. Note that the element of the splitted is the collections of samples from Classes 0 and Classes k.
`newY`	A numeric number denotes the value of the labels from 0 to K which is the number of categories

Details

logit_model fits the splitted data by using the the individualized binary logistic regression according to the value of newY. Because we use use Class 0 as the baseline for modeling the probability ratio of Class k to Class 0 by fitting K individual logistic models, if newY equal to 0, it means we need fit all elements of the splitted data. Otherwise, we only fit the samples from class 0 and class newY.

Value

`beta_mat`	a matrix contains the estimated coefficient. Note that the beta_mat is a n * p matrix which n is the number of the explanatory variables and p+1 is the number of categories

References

Li, J., Chen, Z., Wang, Z., & Chang, Y. I. (2020). Active learning in multiple-class classification problems via individualized binary models. *Computational Statistics & Data Analysis*, 145, 106911. doi:10.1016/j.csda.2020.106911

See Also

`logit_model_ord` for ordinal case.

Examples

```
## For an example, see example(seq_cat_model)
```

`logit_model_ord` *the individualized binary logistic regression for ordinal response data.*

Description

`logit_model_ord` fit the ordinal data by the individualized binary logistic regression

Usage

```
logit_model_ord(splitted, newY, beta_mat)
```

Arguments

`splitted`	A list containing the datasets which we will use in the categorical case. Note that the element of the splitted is the collections of samples from Classes 0 and Classes k.
`newY`	a numeric number denotes the value of the labels from 0 to K which is the number of categories
`beta_mat`	the initial estimate for the coefficient. Note that the values may be not accurate.

Details

logit_model_ord fits the splitted data by using the the individualized binary logistic regression according to the value of newY. Under the ordinal case, we don't use the all training samples. Instead, we use two consecutive subgroups, such as Classes k - 1 and k , at a time for each individual model. Hence, we need fit the model acrroding to the value of newY. param splitted a list containing the datasets which we will use in the cordinl case. Note that the element of the splitted is the collections of samples from Classes 0k - 1and Classes k.

Value

`beta_mat`	a matrix contains the estimated coefficient. Note that the beta_mat is a n * p matrix which n is the number of the explanatory variables and p+1 is the number of categories

References

Li, J., Chen, Z., Wang, Z., & Chang, Y. I. (2020). Active learning in multiple-class classification problems via individualized binary models. *Computational Statistics & Data Analysis*, 145, 106911. doi:10.1016/j.csda.2020.106911

See Also

`logit_model` for categorical case.

Examples

```
## For an example, see example(seq_ord_model)
```

`print.seqbin` *Print the results by the binary logistic regression model*

Description

`print.seqbin` print the result of the binary logistic regression model used by the method of adaptive shrinkage estimate.

Usage

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

Arguments

`x`	A variable of type `seqbin`
`...`	Additional variables to be transferred to functions

Details

This function is used to present results in a concise way. If we select enough samples that satisfy the stopping criterion, then we show several messages to report the conclusion including the length of fixed size confidence set, the number of samples we choose, the value of coefficient and the time have elapsed. Otherwise, the sample selection process is failed. We need to reduce the length of fixed size confidence set because the smaller the dlen, the larger the sample size we need.

Value

print.seqbin returns several messages to show the conclusion.

`print.seqGEE` *Print the results by the generalized estimating equations.*

Description

`print.seqGEE` print the result of the logistic regression model used by the method of adaptive shrinkage estimate.

Usage

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

Arguments

`x`	A variable of type `seqGEE`
`...`	Additional variables to be transferred to functions

Details

This function is used to present results in a concise way. If we select enough samples that satisfy the stopping criterion, then we show several messages to report the conclusion including the length of fixed size confidence set, the number of samples we choose, the value of coefficient and the index of the non zero coefficient

Value

print.seqGEE returns several messages to show the conclusion.

print.seqmulti *Print the results by the multi-logistic regression model*

Description

print.seqmulti print the result of the multi-logistic regression model

Usage

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

Arguments

x	A variable of type seqmulti
...	Additional variables to be transferred to functions

Details

This function is used to present results in a concise way. If we select enough samples that satisfy the stopping criterion, then we show several messages to report the conclusion including the length of fixed size confidence set, the number of samples we choose and the value of coefficient.

Value

print.seqmulti returns several messages to show the conclusion.

QIC *Calculate quasi-likelihood under the independence model criterion (QIC) for Generalized Estimating Equations.*

Description

Select the optimal model according to the QIC criterion

Usage

```
QIC(y, X, id, beta, nonZeroSet, rho, linkv, corstrv)
```

Arguments

`y`	A matrix containing current response variable
`X`	A data frame containing the covariate for the current samples
`id`	The id for each subject in the X
`beta`	The paramters that we estimate when we use the current samples
`nonZeroSet`	The set of the index of the non zero coefficient
`rho`	A numeric number indicating the estimate of correlation coefficient
`linkv`	A specification for the model link function.
`corstrv`	A character string specifying the correlation structure. The following are permitted: "independence", "exchangeable" and "ar1".

Details

QIC calculates the value of the quasi-likelihood under the independence model criterion for Generalized Estimating Equations. The QIC criterion is actually a generalization of the AIC criterion in the statistical inference of parameters in the longitudinal data analysis framework.

Value

a value indicating how well the model fits

`seq_bin_model` *The sequential logistic regression model for binary classification problem.*

Description

`seq_bin_model` estimates the the effective variables and chooses the subjects sequentially by the logistic regression model for the binary classification case with adaptive shrinkage estimate method.

Usage

```
seq_bin_model(startnum, data.clust, xfix, yfix, d = 0.5,
  criterion = "BIC", pho = 0.05, ptarget = 0.5)
```

Arguments

`startnum`	The initial number of subjects from original dataset.
`data.clust`	Large list obtained through k-means clustering. The samples of the element(data.clust[[1]]) in the data.clust is closer to each other compared to another element.
`xfix`	A dataframe that each row is a sample,each column represents an independent variable. The sample has the minimum variance from each cluster of the data.clust to represent the all samples for the corresponding cluster.

yfix	Numeric vector consists of 0 or 1. The length of yfix must be the same as the xfix.
d	A numeric number specifying the length of the fixed size confidence set for our model. Note that the smaller the d, the larger the sample size and the longer the time costs. The default value is 0.5.
criterion	A character string that determines the model selection criterion to be used, matching one of 'BIC' or 'AIC. The default value is 'BIC'.
pho	A numeric number used in subject selection according to the D-optimality. That is, select the first (rho * length(data)) data from the unlabeled data set and add it to the uncertainty set. The default value is 0.05.
ptarget	A numeric number that help to choose the samples. The default value is 0.5

Details

seq_bin_model is a binary logistic regression model that estimetes the effective variables and determines the samples sequentially from original training data set using adaptive shrinkage estimate given the fixed size confidence set. It's a sequential method that we select sample one by one from data pool. Once it stops, it means we select the enough samples that satisfy the stopping criterion and we can conclude which are the effective variables and its corresponding values and the number of the samples we select.

Value

a list containing the following components

d	the length of the fixed size confidence set that we specify
n	the current sample size when the stopping criterion is satisfied
is_stopped	the label of sequential iterations stop or not. When the value of is_stopped is 1, it means the iteration stops
beta_est	the parameters that we estimate when the the iteration is finished
cov	the covariance matrix between the estimated parameters

References

Wang Z, Kwon Y, Chang YcI (2019). Active learning for binary classification with variable selection. arXiv preprint arXiv:1901.10079.

See Also

seq_GEE_model for generalized estimating equations case

seq_bin_model for binary classification case

seq_ord_model for ordinal case.

Examples

```
# generate the toy example. You should remove '#' to
# run the following command.
# library(doMC)
# registerDoMC(9)
# library(foreach)
beta <- c(-1,1,0,0)
N <- 10000
nclass <- 1000
seed <- 123
data  <- gen_bin_data(beta,N,nclass,seed)
xfix <- data[['X']]
yfix <- data[['y']]
data.clust <- data[['data.clust']]
startnum <- 24
d <- 0.75

# use seq_bin_model to binary classification problem. You can remove '#' to
# run the command.
# results <- seq_bin_model(startnum, data.clust, xfix, yfix, d,
#                          criterion = "BIC", pho = 0.05, ptarget = 0.5)
```

`seq_cat_model` *The sequential logistic regression model for multi-classification problem under the categorical case.*

Description

`seq_cat_model` chooses the subjects sequentially by the logistic regression model for the categorical case.

Usage

```
seq_cat_model(labeled_ids, unlabeled_ids, splitted, newY, train, data,
  d = 0.8, adaptive = "random")
```

Arguments

`labeled_ids`	A numeric vector for the unique identification of the labeled dataset.
`unlabeled_ids`	A numeric vector for the unique identification of the unlabeled dataset.
`splitted`	A list containing the datasets which we will use in the categorical case. Note that the element of the splitted is the collections of samples from Classes 0 and Classes k.
`newY`	A numeric number denotes the value of the labels from 0 to K which is the number of categories.
`train`	A matrix for the labeled samples. Note that the indices of the samples in the train dataset is the same as the labeled_ids.

`data`	A matrix denotes all the data including the labeled samples and the unlabeled samples. Note that the first column of the dataset is the response variable, that's the labels and the rest is the explanatory variables.
`d`	A numeric number specifying the length of the fixed size confidence set for our model. The default value is 0.8.
`adaptive`	A character string that determines the sample selection criterion to be used, matching one of 'random' or 'A_optimal The default value is 'random'.

Details

seq_cat_model is a multinomial logistic regression model that estimate the coefficient of the explanatory variables and determines the samples sequentially from original training data set given the fixed size confidence set under the categorical case. Note that there are two methods to select the samples. One sampling method is random sampling while another is the A-optimality criterion which seeks to minimize the trace of the inverse of the information matrix. In addition, we will use the special model: the individualized binary logistic regression. We will use the specific model to only fit two subgroups of the all dataset and get the estimated coefficient and decide whether to stop sampling. If it shows that we need to continue, we will use one of the samepling method above to pick the sample. Note that if the method is A-optimality, we will pick the most informative subjects.

Value

a list containing the following components

`d`	the length of the fixed size confidence set that we specify
`n`	the current sample size when the stopping criterion is satisfied
`is_stopped`	the label of sequential iterations stop or not. When the value of is_stopped is TRUE, it means the iteration stops
`beta_est`	the estimated coeffificent when the criterion is safisfied
`cov`	the covariance matrix between the estimated parameters
`adaptive`	the sample selection criterion we used

References

Li, J., Chen, Z., Wang, Z., & Chang, Y. I. (2020). Active learning in multiple-class classification problems via individualized binary models. *Computational Statistics & Data Analysis*, 145, 106911. doi:10.1016/j.csda.2020.106911

See Also

`seq_GEE_model` for generalized estimating equations case

`seq_bin_model` for binary classification case

`seq_ord_model` for ordinal case.

Examples

```
# generate the toy example
beta <- matrix(c(1,2,1,-1,1,2), ncol=2)
res <-  gen_multi_data(beta, N = 10000, type = 'cat', test_ratio = 0.3)
train_id <- res$train_id
train <- res$train
test <- res$test
res <- init_multi_data(train_id, train, init_N = 300, type = 'cat')
splitted <- res$splitted
train <- res$train
newY <- res$newY
labeled_ids <- res$labeled_ids
unlabeled_ids <- res$unlabeled_ids
data <- res$data

# use seq_cat_model to multi-classification problem under the categorical case.
# You can remove '#' to run the command.
# start_time <- Sys.time()
# logitA_cat <- seq_cat_model(labeled_ids, unlabeled_ids, splitted, newY,
#                              train, data, d = 0.5, adaptive = "A_optimal")
# logitA_cat$time <- as.numeric(Sys.time() - start_time, units = "mins")
# print(logitA_cat)
```

---

seq_GEE_model — *The The sequential method for generalized estimating equations case.*

Description

seq_GEE_model estimates the the effective variables and chooses the subjects sequentially by the generalized estimating equations with adaptive shrinkage estimate method.

Usage

```
seq_GEE_model(formula, data = list(), clusterID, data_pool = list(),
  clusterID_pool, strategy, d = 0.4, family = stats::gaussian(link =
  "identity"), corstr = "independence", contrasts = NULL, ...)
```

Arguments

formula	An object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
data	A data frame containing the initial random samples to obtain the initial estimate of the coefficient. Note that the first column of the data frame is the response variable, and the rest is the explanatory variables.
clusterID	The id for each subject in the initial samples. Note that the subjects in the same cluster will have identical id.

`data_pool`	A data frame containing all the random samples which we will choose subject from. The first column of the data frame is the response variable, and the rest is the explanatory variables.
`clusterID_pool`	The id for each subject in the data_pool. Note that the subjects in the same cluster will have identical id.
`strategy`	A character string that determines the sample selection criterion to be used, matching one of 'random' or 'D_optimal. The default value is 'D_optimal'.
`d`	A numeric number specifying the length of the fixed size confidence set for our model. The default value is 0.4.
`family`	A description of the error distribution and link function to be used in the model. See family for details of `family` functions. Matching one of 'gaussian' or 'binomial'
`corstr`	A character string specifying the correlation structure. The following are permitted: "independence", "exchangeable" and "ar1".
`contrasts`	An optional list. See the contrasts.arg of `model.matrix.default`.
`...`	Further arguments passed to or from other methods.

Details

seq_GEE_model fits the clustered data sequentially by generalized estimating equations with adaptive shrinkage estimate. It can detect the effective variables which have the impact on the response and choose the most representative sample point at the same time. Specifically, we fit a initial sample data and determine if the stop condition is reached. If not, we will select the most informative subjects by some criterion. Iteration stops once it meets our requirements.

Value

a list containing the following components

`d`	the length of the fixed size confidence set that we specify
`n`	the current sample size when the stopping criterion is satisfied
`is_stopped`	the label of sequential iterations stop or not. When the value of is_stopped is TRUE, it means the iteration stops
`beta_est`	the parameters that we estimate when the the iteration is finished
`cov`	the covariance matrix between the estimated parameters
`rho`	estimate of correlation coefficient
`nonZeroIdx`	the index of the non zero coefficient
`corstr`	the correlation structure. The following are permitted: "independence", "exchangeable" and "ar1"
`family`	a description of the error distribution and link function to be used in the model

References

Chen, Z., Wang, Z., & Chang, Y. I. (2019). Sequential adaptive variables and subject selection for GEE methods. *Biometrics*. doi:10.1111/biom.13160

See Also

seq_cat_model for categorical case

seq_bin_model for binary classification case

seq_ord_model for ordinal case.

Examples

```
# generate the toy example
data <- gen_GEE_data(numClusters = 75, clusterSize = 5,
                    clusterCorstr = 'ar1', clusterRho = 0.3,
                    beta = c(1, -1.1, 1.5, -2, rep(0, 50)), family = gaussian(),
                    intercept = TRUE, xCorstr = 'ar1',
                    xCorRho = 0.5, xVariance = 0.2)
df <- data.frame(y = data$y, data$x)
clusterID <- data$clusterID
pool <- gen_GEE_data(numClusters = 8000, clusterSize = 5,
                     clusterCorstr = 'ar1', clusterRho = 0.3,
                     beta = c(1, -1.1, 1.5, -2, rep(0, 50)), family = gaussian(),
                     intercept = TRUE, xCorstr = 'ar1',
                     xCorRho = 0.5, xVariance = 0.2)
df_pool <- data.frame(y = pool$y, pool$x)
clusterID_pool <- pool$clusterID
d<- 0.25

# use seq_GEE_model to generalized estimating equations case.
# You can remove #' to run the command.
# seqRes.ASED <- seq_GEE_model(y ~ .-1, data = df, clusterID = clusterID,
#                              data_pool = df_pool,  clusterID_pool = clusterID_pool,
#                       strategy = "D-optimal",  d = d, family = gaussian(), corstr = 'ar1')
```

`seq_ord_model` *The sequential logistic regression model for multi-classification problem under the ordinal case.*

Description

`seq_ord_model` chooses the subjects sequentially by the logistic regression model for ordinal case

Usage

```
seq_ord_model(labeled_ids, unlabeled_ids, splitted, newY, train, data,
  d = 0.8, adaptive = "random")
```

Arguments

labeled_ids	A numeric vector for the unique identification of the labeled dataset
unlabeled_ids	A numeric vector for the unique identification of the unlabeled dataset

splitted	A list containing the datasets which we will use in the cordinl case. Note that the element of the data_split is the samples from Classes k-1and Classes k
newY	A numeric number denotes the value of the labels from 0 to K which is the number of categories
train	A matrix for the labeled samples. Note that the indices of the samples in the train dataset is the same as the labeled_ids
data	A matrix denotes all the data including the labeled samples and the unlabeled samples. Note that the first column of the dataset is the response variable, that's the labels and the rest is the explanatory variables.
d	A numeric number specifying the length of the fixed size confidence set for our model. The default value is 0.8.
adaptive	A character string that determines the sample selection criterion to be used, matching one of 'random' or 'A_optimal'. The default value is 'random'.

Details

The `seq_ord_model` function and `seq_cat_model` function are very similar. seq_ord_model is also a multinomial logistic regression model but under the ordinal case that estimate the coefficient variables and determines the samples given the fixed size confidence set. seq_ord_model selects the sample in the same way as seq_cat_model: both are two methods. The details about the selecting methoed in seq_ord_model please refer to the `seq_cat_model` function.

Value

a list containing the following components

d	the length of the fixed size confidence set that we specify
n	the current sample size when the stopping criterion is satisfied
is_stopped	the label of sequential iterations stop or not. When the value of is_stopped is TRUE, it means the iteration stops
beta_est	the estimated coeffificent when the criterion is safisfied
cov	the covariance matrix between the estimated parameters
adaptive	the sample selection criterion we used

References

Li, J., Chen, Z., Wang, Z., & Chang, Y. I. (2020). Active learning in multiple-class classification problems via individualized binary models. *Computational Statistics & Data Analysis*, 145, 106911. doi:10.1016/j.csda.2020.106911

See Also

`seq_cat_model` for categorical case

`seq_bin_model` for binary classification case

`seq_GEE_model` for generalized estimating equations case.

Examples

```
# generate the toy example
beta <- matrix(c(1,2,1,-1,1,2), ncol=2)
res <-  gen_multi_data(beta, N = 10000, type = 'ord', test_ratio = 0.3)
train_id <- res$train_id
train <- res$train
test <- res$test
res <- init_multi_data(train_id, train, init_N = 300, type = 'ord')
splitted <- res$splitted
train <- res$train
newY <- res$newY
labeled_ids <- res$labeled_ids
unlabeled_ids <- res$unlabeled_ids
data <- res$data

# use seq_ord_model to multi-classification problem under the ordinal case.
# You can remove '#' to run the command.
# start_time <- Sys.time()
# logitA_ord <- seq_ord_model(labeled_ids, unlabeled_ids, splitted, newY,
#                              train, data, d = 0.5, adaptive = "A_optimal")
# logitA_ord$time <- as.numeric(Sys.time() - start_time, units = "mins")
# print(logitA_ord)
```

---

update_data_cat — *Add the new sample into labeled dataset from unlabeled dataset for the categorical case*

Description

`update_data_cat` selects the sample to the labeled dataset according to it's index

Usage

```
update_data_cat(ind, splitted, data, train, labeled_ids, unlabeled_ids)
```

Arguments

`ind`	A numeric value denotes the index of selected sample.
`splitted`	A list containing the datasets which we will use in the categorical case. Note that the element of the splitted is the collections of samples from Classes 0 and Classes k.
`data`	A matrix denotes all the data including the labeled samples and the unlabeled samples. Note that the first column of the dataset is the response variable, that's the labels and the rest is the explanatory variables.
`train`	A matrix for the labeled samples.
`labeled_ids`	A numeric vector for the unique identification of the labeled dataset
`unlabeled_ids`	A numeric vector for the unique identification of the unlabeled dataset

Details

update_data_cat chooses the sample based on the index from all the training dataset if the data has no ordinal relation. Specifically, we remove the index of the choosed sample from the unlabeled dataset and add the index to the labeled dataset. Then, combine the selected sample with the existing training data set.

Value

splitted	a list containing the datasets which we add a new sample into it
newY	the label of the choosed sample
train	the dataset used for training the model after adding the new sample
labeled_ids	the id of the labeled dataset after updating
unlabeled_ids	the id of the unlabeled dataset after updating

References

Li, J., Chen, Z., Wang, Z., & Chang, Y. I. (2020). Active learning in multiple-class classification problems via individualized binary models. *Computational Statistics & Data Analysis*, 145, 106911. doi:10.1016/j.csda.2020.106911

See Also

update_data_ord

Examples

```
## For an example, see example(seq_cat_model)
```

---

update_data_ord *Add the new sample into labeled dataset from unlabeled dataset for the ordinal case*

Description

update_data_ord selects the sample to the labeled dataset according to it's index

Usage

```
update_data_ord(ind, splitted, data, train, labeled_ids, unlabeled_ids)
```

Arguments

`ind`	A numeric value denotes the index of selected sample.
`splitted`	A list containing the datasets which we will use in the cordinl case. Note that the element of the data_split is the samples from Classes k-1and Classes k
`data`	A matrix denotes all the data including the labeled samples and the unlabeled samples. Note that the first column of the dataset is the response variable, that's the labels and the rest is the explanatory variables.
`train`	A matrix for the labeled samples.
`labeled_ids`	The unique identification of the labeled dataset
`unlabeled_ids`	The unique identification of the unlabeled dataset

Details

update_data_ord chooses the sample based on the index from all the training ordinal dataset. We record the corresponding label of the selected sample and update the data of the unlabeled dataset and the labeled dataset. Specifically, we remove the index of the choosed sample from the unlabeled dataset and add the sample to the labeled dataset.

Value

`splitted`	a list containing the new datasets which we add a new sample into it
`newY`	the label of the choosed sample
`train`	the dataset used for training the model after adding the new sample
`labeled_ids`	the id of the labeled dataset after updating
`unlabeled_ids`	the id of the unlabeled dataset after updating

References

Li, J., Chen, Z., Wang, Z., & Chang, Y. I. (2020). Active learning in multiple-class classification problems via individualized binary models. *Computational Statistics & Data Analysis*, 145, 106911. doi:10.1016/j.csda.2020.106911

See Also

`update_data_cat`

Examples

```
## For an example, see example(seq_ord_model)
```

Index