Package 'HMTL'

July 21, 2025

Type Package

Title Heterogeneous Multi-Task Feature Learning

Version 0.1.0

Description The heterogeneous multi-task feature learning is a data integration method to conduct joint feature selection across multiple related data sets with different distributions. The algorithm can combine different types of learning tasks, including linear regression, Huber regression, adaptive Huber, and logistic regression. The modified version of Bayesian Information Criterion (BIC) is produced to measure the model performance. Package is based on Yuan Zhong, Wei Xu, and Xin Gao (2022) <https://www.fields.utoronto.ca/talk-media/1/53/65/slides.pdf>.

Depends R (>= 3.5.0), stats, graphics, Matrix, pROC

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

NeedsCompilation no

Author Yuan Zhong [aut, cre],
Wei Xu [aut],
Xin Gao [aut]

Maintainer Yuan Zhong <aqua.zhong@gmail.com>

Repository CRAN

Date/Publication 2023-05-04 19:20:02 UTC

Contents

HMTL-package   *Heterogeneous Multi-task Feature Learning*

Description

HMTL package implements the block-wise sparse estimation by grouping the coefficients of related predictors across multiple tasks. The tasks can be either regression, Huber regression, adaptive Huber regression, and logistic regression, which provide a wide variety of data types for the integration. The robust methods, such as the Huber regression and adaptive Huber regression, can deal with outlier contamination based on Sun, Q., Zhou, W.-X. and Fan, J. (2020), and Wang, L., Zheng, C., Zhou, W. and Zhou, W.-X. (2021). The model selection applies the modified form of Bayesian information criterion to measure the mdoel performance, which has similar formulation as previous work developed by Gao, X., and Carroll, R. J., (2017).

Details

In the context of multi-task learning, there are K different data sets obtained from K related sources. The data sets can be modeled by different types of learning tasks based on the data distributions. Let the candidate features be denoted as $\{M_1, M_2, ..., M_j, ..., M_p\}$. When the integrated data sets have different measurements, we assume the predictors to share some similarities. For example, the jth predictors collected as $M_j = (X_{1j}, X_{2j}, \cdots, X_{Kj})$ in the table below represent the same type of feature in all related studies. In some cases, the tasks can share same set of predictor, then $X_{1j} = X_{2j} = \cdots = X_{Kj}$.

Tasks	Formula	M_1	M_2	...	M_j	...	M_p
1	$y_1 : g_1(\mu_1) \sim$	$x_{11}\theta_{11}+$	$x_{12}\theta_{12}+$	...	$x_{1j}\theta_{1j}+$	...	$x_{1p}\theta_{1p}$
2	$y_2 : g_2(\mu_2) \sim$	$x_{21}\theta_{21}+$	$x_{22}\theta_{22}+$	...	$x_{2j}\theta_{2j}+$	...	$x_{2p}\theta_{2p}$
...							
K	$y_K : g_K(\mu_K) \sim$	$x_{K1}\theta_{K1}+$	$x_{K2}\theta_{K2}+$	...	$x_{Kj}\theta_{Kj}+$	...	$x_{Kp}\theta_{Kp}$

The coefficients can be grouped as the vector $\theta^{(j)}$ for the feature M_j.

Platforms	$\boldsymbol{M_j}$	$\boldsymbol{\theta^{(j)}}$
1	x_{1j}	θ_{1j}
2	x_{2j}	θ_{2j}
...	...	...
k	x_{Kj}	θ_{Kj}

The heterogeneous multi-task feature learning HMTL can select significant features through the overall objective function:

$$Q(\theta) = \mathcal{L}(\theta) + \mathcal{R}(\theta).$$

The loss function is defined as $\mathcal{L}(\theta) = \sum_{k=1}^{K} w_k \ell_k(\theta_k)$, which can be the composite quasi-likelihood or the composite form of (adaptive) Huber loss with additional robustification parameter τ_k. The penalty function is the mixed $\ell_{2,1}$ regularization, such that $\mathcal{R}(\theta) = \lambda \sum_{j=1}^{p} (\sum_{k=1}^{K} \theta_{kj}^2)^{1/2}$.

This package also contains functions to provide the Bayesian information criterion:

$$BIC(s) = 2\mathcal{L}_s(\hat{\theta}) + d_s^* \gamma_n$$

with $\mathcal{L}_s(\hat{\theta})$ denoting the composite quasi-likelihood or adaptive Huber loss, d_s^* measuring the model complexity and γ_n being the penalty on the model complexity.

In this package, the function `MTL_reg` deals with regression tasks, which can be outlier contaminated. The function `MTL_class` is applied to model multiple classification tasks, and the function `MTL_hetero` can integrate different types of tasks together.

Author(s)

Yuan Zhong, Wei Xu, and Xin Gao

Maintainer: Yuan Zhong <aqua.zhong@gmail.com>

References

Zhong, Y., Xu, W., and Gao X., (2023) Heterogeneous multi-task feature learning with mixed $\ell_{2,1}$ regularization. Submitted

Zhong, Y., Xu, W., and Gao X., (2023) Robust Multi-task Feature Learning. Submitted

Gao, X., and Carroll, R. J., (2017) Data integration with high dimensionality. Biometrika, 104, 2, pp. 251-272

Huber, P. J. (1964). Robust estimation of a location parameter. Ann. Math. Statist., 35, 73–101.

Sun, Q., Zhou, W.-X. and Fan, J. (2020). Adaptive Huber regression. J. Amer. Statist. Assoc., 115, 254-265.

Wang, L., Zheng, C., Zhou, W. and Zhou, W.-X. (2021). A new principle for tuning-free Huber regression. Stat. Sinica, 31, 2153-2177.

---

mockdata   *Mock Gene Data*

Description

This data set is a mock version of two related research outcomes.

Usage

```
data("mockdata1")
data("mockdata2")
```

Format

The `mockdata1` contains all predictor variables for four data sets, and the `mockdata2` is the list of four columns of response variables. The sample sizes are $n_1 = 500, n_2 = 250, n_3 = 250$, and $n_4 = 250$.

The response varibles are heterogeneous, such that the first two columns are continuous data and last two are binary data. The predictors of each data matrix in `mockdata1` have 500 columns of variables, and the columns in different data matrices are matched. For example, the first column in the first data matrix represents the same type of feature as the first columns in other three data matrices. Among all canadidate predictors, the response variables are set to be associated with the first 22 columns, and the remaining columns are not important predictors.

Details

This data set is used as an example to implement functions in the package.

MTL_class   *Multiple Classification Task Feature Learning*

Description

`MTL_class` conducts multi-tasks feature learning to the learning tasks with binary response variables, namely logistic regression. The penalty function applies a mixed $\ell_{2,1}$ norm to combine regression coefficients of predictor shared across all tasks.

Usage

```
MTL_class(
  y,
  x,
  lambda,
  Kn,
  p,
  n,
  beta = 0.1,
  import_w = 1,
  tol = 0.05,
  max_iter = 100,
  Complete = "True",
  diagnostics = FALSE,
  gamma = 1,
  alpha = 1
)
```

Arguments

y	List. A list of binary responses vectors for all tasks.
x	List. Listing matrices of the predictors for all tasks align with the same order as in y.
lambda	Numeric. The penalty parameter used for block-wise regularization ($\ell_{2,1}$ norm).
Kn	Numeric. The number of tasks with binary responses.
p	Numeric. The number of features.
n	Numeric or vector. If only one numeric value is provided, equal sample size will be assumed for each task. If a vector is provided, then the elements are the sample sizes for all tasks.
beta	(**optional**). Numeric or matrix. An initial value or matrix of values p by K for the estimation. The default value is 0.1.
import_w	Numeric or vector. The weights assigned to different tasks. An equal weight is set as the default.
tol	(**optional**). Numeric. The tolerance level of optimation.
max_iter	(**optional**). Numeric. The maximum number of iteration steps.
Complete	Logic input. If the predictors in each task are all measured, set 'Complete == TRUE'; If some predictors in some but not all task are all measured, set 'Complete == FALSE', and the missing values are imputed by column mean. The adjustment weights will be assigned based on the completeness of the predictors.
diagnostics	Logic input. If 'diagnostics == TRUE', the function provides Bayesian information criterion, and the selected model performance is evalued by the MSE and MAE for tasks with continuous response and the AUC and deviance for tasks with binary responses.
gamma	(**optional**). Numeric. Step size for each inner iteration. The default is equal to 1.
alpha	(**optional**). Numeric. A tuning parameter for BIC penalty. The default is equal to 1.

Value

A list including the following terms will be returned:

beta A p by K matrix of estimated sparse parameters.

Task type The models used in each task.

Task weights The weights assigned to each task.

Selected_List The index of non-zero parameters.

If 'diagnostics = TRUE', the following terms will be returned:

Bayesian_Information Table of the information criterion: Composite likelihood, Degree of freedom, and (peudo or robust) Bayesian informtion criterion.

Class_Perform Table of the model performance for classification tasks: the area under ROC curve (AUC), and the deviance (DEV) estimated by 'glm'.

Residuals The residuals for all tasks.

References

Zhong, Y., Xu, W., and Gao X., (2023) Heterogeneous multi-task feature learning with mixed $\ell_{2,1}$ regularization. Submitted

Examples

```
x_class <- list(mockdata1[[3]],mockdata1[[4]])
y_class <- list(mockdata2[[3]],mockdata2[[4]])
model <- MTL_class(y_class,x_class, lambda = 2/11 , Kn = 2, p=500,
                   n = 250 ,gamma = 1, Complete = FALSE, diagnostics = TRUE, alpha = 2)
# Selected non-zero coefficients
model$beta[model$Selected_List,]
# Estimated Pseudo-BIC
model$Bayesian_Information
# Classification accuracy
model$Class_Perform
```

MTL_hetero   *Heterogeneous Multi-task Feature Learning*

Description

`MTL_hetero` conducts multi-tasks feature learning to different types of learning tasks, including linear regression, Huber regression, adaptive Huber, and logistic regression. The penalty function applies a mixed $\ell_{2,1}$ norm to combine regression coefficients of predictor shared across all tasks.

Usage

```
MTL_hetero(
  y,
  x,
  lambda,
  Kn,
  p,
  n,
  beta = 0.1,
  tau = 1.45,
  Cont_Model = "adaptive Huber",
  import_w = 1,
  tol = 0.05,
  max_iter = 100,
  Complete = "True",
  diagnostics = FALSE,
  gamma = 1,
  alpha = 1
)
```

Arguments

y	List. A list of responses vectors for all tasks. The order of the list put the continuous responses before the binary responses.
x	List. Listing matrices of the predictors for all tasks align with the same order as in y.
lambda	Numeric. The penalty parameter used for block-wise regularization ($\ell_{2,1}$ norm).
Kn	Vector of two elements. First element is the number of tasks with continuous responses, and the second element is the number of tasks with binary responses.
p	Numeric. The number of features.
n	Numeric or vector. If only one numeric value is provided, equal sample size will be assumed for each task. If a vector is provided, then the elements are the sample sizes for all tasks.
beta	(**optional**). Numeric or matrix. An initial value or matrix of values p by K for the estimation. The default value is 0.1.
tau	Numeric or vector. The robustification parameter used for methods "Huber regression" or "Adaptive Huber". The default value is 1.45.
Cont_Model	Character("regression", "Huber regression", or "adaptive Huber"). The models used for tasks with continuous responses.
import_w	Numeric or vector. The weights assigned to different tasks. An equal weight is set as the default.
tol	(**optional**). Numeric. The tolerance level of optimation.
max_iter	(**optional**). Numeric. The maximum number of iteration steps.
Complete	Logic input. If the predictors in each task are all measured, set 'Complete == TRUE'; If some predictors in some but not all task are all measured, set 'Complete == FALSE', and the missing values are imputed by column mean. The adjustment weights will be assigned based on the completeness of the predictors.
diagnostics	Logic input. If 'diagnostics == TRUE', the function provides Bayesian information criterion, and the selected model performance is evalued by the MSE and MAE for tasks with continuous response and the AUC and deviance for tasks with binary responses.
gamma	(**optional**). Numeric. Step size for each inner iteration. The default is equal to 1.
alpha	(**optional**). Numeric. A tuning parameter for BIC penalty. The default is equal to 1.

Value

A list including the following terms will be returned:

beta A p by K matrix of estimated sparse parameters.

Task type The models used in each task.

Task weights The weights assigned to each task.

Selected_List The index of non-zero parameters.

If 'diagnostics = TRUE', the following terms will be returned:

`Bayesian_Information` Table of the information criterion: Composite likelihood, Degree of freedom, and (peudo or robust) Bayesian informtion criterion.

`Reg_Error` Table of the model performance for (Huber) regressions: the mean square error (MSE), and the mean absolute error (MAE).

`Class_Perform` Table of the model performance for classification tasks: the area under ROC curve (AUC), and the deviance (DEV) estimated by 'glm'.

`Residuals` The residuals for all tasks.

Note

When penalty parameter is too small, the estimated coefficients may have $p \geq n$. The algorithm can provide the estimated values, but the `diagnostics`'s results will not be given.

References

Zhong, Y., Xu, W., and Gao X., (2023) Heterogeneous multi-task feature learning with mixed $\ell_{2,1}$ regularization. Submitted

Examples

```
model <- MTL_hetero(mockdata2,mockdata1, lambda =2.5, Kn = c(2,2), p=500,n = c(500,250,250,250),
                    gamma = 2, Complete = FALSE, diagnostics = TRUE, alpha = 2)
# Selected non-zero coefficients
model$beta[model$Selected_List,]
# Estimated Pseudo-BIC
model$Bayesian_Information
# Regression error
model$Reg_Error
# Classification accuracy
model$Class_Perform
```

MTL_reg   *Robust Multi-Task Feature Learning*

Description

`MTL_reg` conducts multi-tasks feature learning to the learning tasks with continous response variables, such as the linear regression, Huber regression, adaptive Huber. The adaptive Huber method is based on Sun, Q., Zhou, W.-X. and Fan, J. (2020) and Wang, L., Zheng, C., Zhou, W. and Zhou, W.-X. (2021). The penalty function applies a mixed $\ell_{2,1}$ norm to combine regression coefficients of predictor shared across all tasks. The Huber regression and adaptive Huber regression need the robustification parameter τ_k to strike a balance between the unbiasedness and robustness, and the adaptive method can determine this parameter by a tuning-free principle.

Usage

```
MTL_reg(
  y,
  x,
  lambda,
  Kn,
  p,
  n,
  beta = 0.1,
  tau = 1.45,
  Cont_Model = "adaptive Huber",
  import_w = 1,
  tol = 0.05,
  max_iter = 100,
  Complete = "True",
  diagnostics = FALSE,
  gamma = 1,
  alpha = 1
)
```

Arguments

y	List. A list of continuous responses vectors for all tasks.
x	List. Listing matrices of the predictors for all tasks align with the same order as in y.
lambda	Numeric. The penalty parameter used for block-wise regularization ($\ell_{2,1}$ norm).
Kn	Numeric. The number of tasks with continuous responses.
p	Numeric. The number of features.
n	Numeric or vector. If only one numeric value is provided, equal sample size will be assumed for each task. If a vector is provided, then the elements are the sample sizes for all tasks.
beta	(**optional**). Numeric or matrix. An initial value or matrix of values p by K for the estimation. The default value is 0.1.
tau	Numeric or vector. The robustification parameter used for methods "Huber regression" or "Adaptive Huber". The default value is 1.45.
Cont_Model	Character("regression", "Huber regression", or "adaptive Huber"). The models used for tasks with continuous responses.
import_w	Numeric or vector. The weights assigned to different tasks. An equal weight is set as the default.
tol	(**optional**). Numeric. The tolerance level of optimation.
max_iter	(**optional**). Numeric. The maximum number of iteration steps.
Complete	Logic input. If the predictors in each task are all measured, set 'Complete == TRUE'; If some predictors in some but not all task are all measured, set 'Complete == FALSE', and the missing values are imputed by column mean. The adjustment weights will be assigned based on the completeness of the predictors.

diagnostics	Logic input. If 'diagnostics == TRUE', the function provides Bayesian information criterion, and the selected model performance is evalued by the MSE and MAE for tasks with continuous response and the AUC and deviance for tasks with binary responses.
gamma	(**optional**). Numeric. Step size for each inner iteration. The default is equal to 1.
alpha	(**optional**). Numeric. A tuning parameter for BIC penalty. The default is equal to 1.

Value

A list including the following terms will be returned:

beta A p by K matrix of estimated sparse parameters.

Task type The models used in each task.

Task weights The weights assigned to each task.

Selected_List The index of non-zero parameters.

If 'diagnostics = TRUE', the following terms will be returned:

Bayesian_Information Table of the information criterion: Composite likelihood, Degree of freedom, and (peudo or robust) Bayesian informtion criterion.

Reg_Error Table of the model performance for (Huber) regressions: the mean square error (MSE), and the mean absolute error (MAE).

Residuals The residuals for all tasks.

References

Huber, P. J. (1964). Robust estimation of a location parameter. Ann. Math. Statist., 35, 73–101.

Sun, Q., Zhou, W.-X. and Fan, J. (2020). Adaptive Huber regression. J. Amer. Statist. Assoc., 115, 254-265.

Wang, L., Zheng, C., Zhou, W. and Zhou, W.-X. (2021). A new principle for tuning-free Huber regression. Stat. Sinica, 31, 2153-2177.

Zhong, Y., Xu, W., and Gao X., (2023) Robust Multi-task Feature Learning. Submitted

Examples

```
x_reg <- list(mockdata1[[1]],mockdata1[[2]])
y_reg <- list(mockdata2[[1]],mockdata2[[2]])
model <- MTL_reg(y_reg,x_reg, lambda = 2.5  , Kn = 2, p=500,
            n = c(500,250 ),gamma = 2, Complete = FALSE, diagnostics = TRUE, alpha = 2)

# Selected non-zero coefficients
model$beta[model$Selected_List,]
# Estimated Pseudo-BIC
model$Bayesian_Information
# Regression error
model$Reg_Error
```

---

plot_HMTL *Plot diagram of the information criterion vs. penalty parameters*

Description

Plot a diagram to illustrate the change of Bayesian information criterion with different penalty paameters for model selection

Usage

```
plot_HMTL(x)
```

Arguments

x   Object of class "Model Selection" created by Selection_HMTL

Value

X axis represents the value of penalty parameters, and y axis represents the estimated values of the composite likelihood, degree of freedom for model complexity, and the (robust) Bayesian information criterion.

Selection_HMTL *Model Selection for Multi-task Feature Learning based on Bayesian Information Criterion (BIC)*

Description

Selection_HMTL can be used to search the optimal candidate model based on Bayesian Information Criterion (BIC).

Usage

```
Selection_HMTL(
  y,
  x,
  lambda,
  Kn,
  p,
  n,
  beta = 0.1,
  tau = 1.45,
  Cont_Model = "adaptive Huber",
  type = "Heterogeneity",
  import_w = 1,
  tol = 0.05,
```

```
    max_iter = 100,
    Complete = "True",
    diagnostics = FALSE,
    gamma = 1,
    alpha = 1
)
```

Arguments

y	List. A list of responses vectors for all tasks. The order of the list put the continuous responses before the binary responses.
x	List. Listing matrices of the predictors for all tasks align with the same order as in y.
lambda	Numeric. The penalty parameter used for block-wise regularization ($\ell_{2,1}$ norm).
Kn	Vector of two elements. First element is the number of tasks with continuous responses, and the second element is the number of tasks with binary responses.
p	Numeric. The number of features.
n	Numeric or vector. If only one numeric value is provided, equal sample size will be assumed for each task. If a vector is provided, then the elements are the sample sizes for all tasks.
beta	(**optional**). Numeric or matrix. An initial value or matrix of values p by K for the estimation. The default value is 0.1.
tau	Numeric or vector. The robustification parameter used for methods "Huber regression" or "Adaptive Huber". The default value is 1.45.
Cont_Model	Character("regression", "Huber regression", or "adaptive Huber"). The models used for tasks with continuous responses.
type	Character("Heterogeneity", "Continuous" or "Binary"). type = "Heterogeneity" used for MTL_hetero; type = "Continuous" used for MTL_reg; and type = "Binary" used for MTL_class.
import_w	Numeric or vector. The weights assigned to different tasks. An equal weight is set as the default.
tol	(**optional**). Numeric. The tolerance level of optimation.
max_iter	(**optional**). Numeric. The maximum number of iteration steps.
Complete	Logic input. If the predictors in each task are all measured, set 'Complete == TRUE'; If some predictors in some but not all task are all measured, set 'Complete == FALSE', and the missing values are imputed by column mean. The adjustment weights will be assigned based on the completeness of the predictors.
diagnostics	Logic input. If 'diagnostics == TRUE', the function provides Bayesian information criterion, and the selected model performance is evalued by the MSE and MAE for tasks with continuous response and the AUC and deviance for tasks with binary responses.
gamma	(**optional**). Numeric. Step size for each inner iteration. The default is equal to 1.
alpha	(**optional**). Numeric. A tuning parameter for BIC penalty. The default is equal to 1.

Details

The Bayesian information criterion is given by

$$BIC(s) = 2\mathcal{L}_s(\hat{\theta}) + d_s^* \gamma_n,$$

where $\hat{\theta}$ is the estimated coefficients and s is denoted the selected support set of $\hat{\theta}$. In addition, $\mathcal{L}_s(\hat{\theta})$ denoted the estimated composite quasi-likelihood or adaptive Huber loss evaluated as the $\hat{\theta}$. The degree of freedom d_s^* can measure the model complexity, which is estimated by $tr(H_s^{-1}(\hat{\theta})J_s(\hat{\theta}))$. The sensitivity matrix and specificity matrix can be given by $H(\theta) = E(\nabla^2 \mathcal{L}(\theta))$ and $J(\theta) = -Cov(\nabla \mathcal{L}(\theta))$. The penalty term γ_n can be defined by users.

Value

A table of Bayesian Information Criterion (BIC)

lambda   A list of penalty parameters.

Compo_likelihood   Sum of empirical loss functions estimated based on the selected parameters .

Degree of freedom   Penalty component based on the selected parameters.

Info criterion   Bayesian Information Criterion (BIC): robust BIC or pseudo BIC.

References

Y. Zhong, W. Xu, and X. Gao (2023) Robust Multi-task Feature Learning. Submitted

Gao, X and Carroll, R. J. (2017) Data integration with high dimensionality. Biometrika, 104, 2, pp. 251-272

Examples

```
lambda <- c(1.2, 2.0, 2.3, 2.8)
cv.mtl <- Selection_HMTL(mockdata2,mockdata1, lambda =lambda, Kn = c(2,2), p=500,
          n = c(500,250,250,250),gamma = 2, Complete = FALSE, diagnostics = TRUE, alpha = 2)
plot_HMTL(cv.mtl)
cv.mtl$Selection_Results
```

Index