Package 'bmsr'

March 3, 2021

Type Package

Title Bayesian multi-source regression

Version 1.0.0

Author Suleiman A. Khan [aut, cre],
Brian White [aut]

Maintainer Suleiman A. Khan <khan.suleiman@gmail.com>

Description The bmsr package implements joint regression from multiple data sources in a Bayesian framework. The package provides implementation for both single-task and multi-task regression. The model is implemented using STAN and interface is provided using R programming language. Options for training the model using both NUTS sampler and variational inference are provided. The package is structured for ease of use and the included demo shows the model execution on real-life as well as simulated datasets.

Citation <To Appear>

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

Biarch true

Depends R (>= 3.4.0)

Imports methods,
Rcpp (>= 0.12.0),
RcppParallel (>= 5.0.1),
rstan (>= 2.18.1),
rstantools (>= 2.1.1)

LinkingTo BH (>= 1.66.0),
Rcpp (>= 0.12.0),
RcppEigen (>= 0.3.3.3.0),
RcppParallel (>= 5.0.1),
rstan (>= 2.18.1),
StanHeaders (>= 2.18.0)

SystemRequirements GNU make

R topics documented:

`bmsr-package` *The 'bmsr' package.*

Description

The bmsr package implements joint regression from multiple data sources in a Bayesian framework. The package provides implementation for both single-task and multi-task regression. The model is implemented using STAN and interface is provided using R programming language. Options for training the model using both NUTS sampler and variational inference are provided. The package is structured for ease of use and the included demo shows the model execution on real-life as well as simulated datasets.

References

Brian S. White, Suleiman A. Khan, Mike J Mason, Muhammad Ammad-ud-din, Swapnil Potdar, Disha Malani, Heikki Kuusanmäki, Brian J. Druker, Caroline A Heckman, Olli Kallioniemi, Stephen E Kurtz, Kimmo Porkka, Cristina E. Tognon, Jeffrey W. Tyner, Tero Aittokallio, Krister Wennerberg, Justin Guinney, *Bayesian multi-source regression and monocyte-associated gene expression predict BCL-2 inhibitor resistance in acute myeloid leukemia*, To Appear, (2021)

`demo_bmsmtr` *demo multi-source multi-task model training and interpretative plots*

Description

`demo_bmsmtr` trains bmsmtr on random multi-source multi-task regression dataset.

Usage

```
demo_bmsmtr(dY = 3)
```

Arguments

dY	is a scaler representing the desired number of dimensions in Y (outputs). dY > 1 refers to multi-task datasets.

Examples

```
#for multi-source multi-task regression
demo_bmsmtr(dY = 3)
```

---

demo_bmsr *demo multi-source model training and interpretative plots*

Description

demo_bmsr trains bmsr on random multi-source regression dataset.

Usage

```
demo_bmsr()
```

Examples

```
#for multi-source regression
demo_bmsr()
```

---

demo_bmsr_lapatinib_gdsc_ccle *demo real data training and result visualization*

Description

demo_bmsr_lapatinib_gdsc_ccle trains bmsr model on real dataset from GDSC and CCLE datasets from the PharmacoGx package. The demo trains a model for lapatinib response predictions.

Usage

```
demo_bmsr_lapatinib_gdsc_ccle()
```

Examples

```
#model lapatanib response from GDSC and CCLE
demo_bmsr_lapatinib_gdsc_ccle()
```

generateSyntheticData *generate multi-source toy data for regression*

Description

generateSyntheticData simualtes random multi-source multi-task regression dataset.

Usage

```
generateSyntheticData(S = 2, nY = c(40, 80), dX = 50, dY = 1, ft = 10)
```

Arguments

S	is a scaler representing the desired number of sources.
nY	is a vector representing the desired number of samples in each source.
dX	is a scaler representing the desired number of dimensions in X (inputs).
dY	is a scaler representing the desired number of dimensions in Y (outputs). dY > 1 refers to multi-task datasets.
ft	is a scaler representing the desired number of active features in X.

Value

A list containing the following elements:

data	which is a list containing the Y and X data matrices,
Beta	the beta parameters used to generate the data.

getBeta.bmsmtr.stan *get beta posterior of the bmsmtr model*

Description

getBeta.bmsmtr.stan extracts the posterior values of source specific beta parameters.

Usage

```
getBeta.bmsmtr.stan(out)
```

Arguments

out	is trained STAN model.

Value

beta matrix containing source specific beta parameters.

getBeta.bmsr.stan *get beta posterior of the bmsr model*

Description

`getBeta.bmsr.stan` extracts the posterior values of source specific beta parameters.

Usage

```
getBeta.bmsr.stan(out)
```

Arguments

`out`	is trained STAN model.

Value

beta matrix containing source specific beta parameters.

getPosterior *baseline function to get posterior*

Description

`getPosterior` extracts the posterior values from mode output.

Usage

```
getPosterior(file = NULL, out)
```

Arguments

`file`	is the stan file name containig the stan code.
`out`	is trained STAN model.

Value

post is a list containing posterior of all model weights.

posterior.bmsmtr.stan *get posterior of the bmsmtr model weights*

Description

getPosterior extracts the posterior values from mode output.

Usage

```
posterior.bmsmtr.stan(out)
```

Arguments

out	is trained STAN model.

Value

post is a list containing following posterior weights.

beta	matrix containing source specific beta parameters,
W	matrix of multi-task parameters,
tau	global scaling factor learned,
sigma	noise parameter.

posterior.bmsr.stan *get posterior of the bmsr model weights*

Description

posterior.bmsr.stan extracts the posterior values from mode output.

Usage

```
posterior.bmsr.stan(out)
```

Arguments

out	is trained STAN model.

Value

post is a list containing following posterior weights.

beta	matrix containing source specific beta parameters,
betaShared	vector of shared beta parameters common for all sources,
tau	globalscaling factor learned,
sigma	noise parameter.

predict.bmsmtr.stan *predict function for bmsmtr model*

Description

predict.bmsmtr.stan predicts the output of bmsmtr model, used as predFunction in predictSTAN.

Usage

```
## S3 method for class 'bmsmtr.stan'
predict(out, xTest, nTest, yN = NULL)
```

Arguments

out	is trained STAN model.
xTest	is a matrix of test data for predicting the outcome. If NULL no prediction is made (default).
nTest	is a vector of length S, containing the number of values in each source. Can contain zero's.
yN	a list containing values used for normalizing the data: (default = NULL).

- cmvector of means with which the data is centered, (0's if data is not centered),
- csvector of standard deviations with which the data is scaled (1's if data is not scaled).

Value

yPred prediction vector of the bmsmtr model.

predict.bmsr.stan *predict function for bmsr model*

Description

predict.bmsr.stan predicts the output of bmsr model, used as predFunction in predictSTAN.

Usage

```
## S3 method for class 'bmsr.stan'
predict(out, xTest, nTest, yN = NULL)
```

Arguments

out	is trained STAN model.
xTest	is a matrix of test data for predicting the outcome. If NULL no prediction is made (default).
nTest	is a vector of length S, containing the number of values in each source. Can contain zero's.
yN	a list containing values used for normalizing the data: (default = NULL). • cmvector of means with which the data is centered (0's if data is not centered), • csvector of standard deviations with which the data is scaled (1's if data is not scaled).

Value

yPred prediction vector of the bmsr model.

---

predictSTAN *predict from a regression STAN model*

Description

predictSTAN predicts the output a stan regression model as defined by the parameter predFunction.

Usage

```
predictSTAN(predFunction, out, xTest, nTest, yN = NULL)
```

Arguments

predFunction	is a function for predicting y's using the outcome of stan run.
xTest	is a matrix of test data for predicting the outcome. If NULL no prediction is made (default).
nTest	is a vector of length S, containing the number of values in each source. Can contain zero's.
yN	a list containing values used for normalizing the data: (default = NULL). • cmvector of means with which the data is centered (0's if data is not centered), • csvector of standard deviations with which the data is scaled (1's if data is not scaled).

Value

yPred prediction vector of the stan model

runSTAN *train a regression STAN model*

Description

runSTAN runs a stan regression model and predicts the values for the test samples.

Usage

```
runSTAN(file, data, opts)
```

Arguments

file	is the stan file containing the stan code.
data	is a list containing the data in the format this stan code accepts.
opts	a list containing opts to run the model: \itemiterinteger identifies number of sampling iterations, \itemseedsvector of integers identifying seeds for running the method, \iteminferencestring identifying the sampling method to use, either of Sampling or VB.

Value

A list containing the following elements:

out	STAN output variable,
runtime	run time of the code.

ztransform *z-transform a training data matrix*

Description

ztransform performs column-wise mean=0 and sd=1 normalization of a matrix for all columns starting from FFSC

Usage

```
ztransform(mat, FFSC, verbose = FALSE)
```

Arguments

mat	matrix to z-transform.
FFSC	the index of first column to standardize. All columns including this and after are normalized.
verbose	print logs or not. Defaults to FALSE.

Value

a list containing the following:

mat	column-wise z-transformed matrix,
cm	vector of column means used for z-transform,
cs	vector of column standard deviations used for z-transform.

---

ztransformTest *z-transform a test data matrix using column means and column standard deviations from training data matrix*

Description

ztransformTest performs column-wise mean=0 and sd=1 normalization of a test matrix for all columns starting from FFSC

Usage

```
ztransformTest(mat, FFSC, cm, cs, verbose = FALSE)
```

Arguments

mat	matrix to z-transform.
FFSC	the index of first column to standardize. All columns including this and after are normalized.
cm	vector of column means used for z-transform.
cs	vector of column standard deviations used for z-transform.
verbose	print logs or not. Defaults to FALSE.

Value

a list containing the following:

mat	column-wise z-transformed matrix,
cm	vector of column means used for z-transform,
cs	vector of column standard deviations used for z-transform.

Index