Package: drglm (via r-universe)

August 30, 2024

Type Package

Title Fitting Linear and Generalized Linear Models in ``Divide and Recombine" Approach to Large Data Sets

Version 1.1

Maintainer Md. Mahadi Hassan Nayem <mhnayem.cu.stat@outlook.com>

Depends R (>= 3.5.0)

Imports nnet, speedglm, stats

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Description To overcome the memory limitations for fitting linear (LM) and Generalized Linear Models (GLMs) to large data sets, this package implements the Divide and Recombine (D&R) strategy. It basically divides the entire large data set into suitable subsets manageable in size and then fits model to each subset. Finally, results from each subset are aggregated to obtain the final estimate. This package also supports fitting GLMs to data sets that cannot fit into memory and provides methods for fitting GLMs under linear regression, binomial regression, Poisson regression, and multinomial logistic regression settings. Respective models are fitted using different D&R strategies as described by: Xi, Lin, and Chen (2009) <doi:10.1109/TKDE.2008.186>, Xi, Lin and Chen (2006) <doi:10.1109/TKDE.2006.196>, Zuo and Li (2018) <doi:10.4236/ojs.2018.81003>, Karim, M.R., Islam, M.A. (2019) <doi:10.1007/978-981-13-9776-9>.

License GPL (>= 3)

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Config/testthat/edition 3

VignetteBuilder knitr

URL https://nayemmh.github.io/drglm/

Repository https://nayemmh.r-universe.dev

RemoteUrl https://github.com/nayemmh/drglm

RemoteRef HEAD

RemoteSha a665499772ddca43348644707f2d97217c9d6e1e

Contents

`big.drglm` *Fitting Linear and Generalized Linear Models to out of the memory data sets in "Divide and Recombine" approach*

Description

Function `big.drglm` aimed to fit GLMs to datasets larger in size that can not be stored in memory. It uses popular divide and recombine technique to handle large data sets efficiently.

Usage

```
big.drglm(data.generator, formula, chunks, family)
```

Arguments

`data.generator`	Using the function `make.data` to initialize the data reading function with the data set path and chunk size, then the data.generate is used directly as data source for the `big.drglm` function.
`formula`	An entity belonging to the "formula" class (or one that can be transformed into that class) represents a symbolic representation of the model that needs to be adjusted. Specifics about how the model is defined can be found in the 'Details' section.
`chunks`	Number of subsets to be divided.
`family`	An explanation of the error distribution that will be implemented in the model.

Value

A Generalized Linear Model is fitted in "Divide & Recombine" approach using preferred number of chunks to data set. A list of model coefficients is estimated using divide and recombine method with the respective standard error of estimates.

Author(s)

MH Nayem

References

- Xi, R., Lin, N., & Chen, Y. (2009). Compression and aggregation for logistic regression analysis in data cubes. IEEE Transactions on Knowledge and Data Engineering, 21(4).
- Chen, Y., Dong, G., Han, J., Pei, J., Wah, B. W., & Wang, J. (2006). Regression cubes with losseless compression and aggregation. IEEE Transactions on Knowledge and Data Engineering, 18(12).
- Zuo, W., & Li, Y. (2018). A New Stochastic Restricted Liu Estimator for the Logistic Regression Model. Open Journal of Statistics, 08(01).
- Karim, M. R., & Islam, M. A. (2019). Reliability and Survival Analysis. In Reliability and Survival Analysis.
- Enea, M. (2009) Fitting Linear Models and Generalized Linear Models with large data sets in R.
- Bates, D. (2009) Technical Report on Least Square Calculations.
- Lumley, T. (2009) *biglm* package documentation.

See Also

drglm, drglm.multinom

Examples

```
# Create a toy dataset
set.seed(123)
# Number of rows to be generated
n <- 10000

# Creating dataset
dataset <- data.frame(
  Var_1 = round(rnorm(n, mean = 50, sd = 10)),
  Var_2 = round(rnorm(n, mean = 7.5, sd = 2.1)),
  Var_3 = as.factor(sample(c("0", "1"), n, replace = TRUE)),
  Var_4 = as.factor(sample(c("0", "1", "2"), n, replace = TRUE)),
  Var_5 = as.factor(sample(0:15, n, replace = TRUE)),
  Var_6 = round(rnorm(n, mean = 60, sd = 5))
)

# Save the dataset to a temporary file
temp_file <- tempfile(fileext = ".csv")
write.csv(dataset, file = temp_file, row.names = FALSE)

# Path to the temporary file
dataset_path <- temp_file
dataset_path  # Display the path to the temporary file

# Initialize the data reading function with the data set path and chunk size
```

```
da <- drglm::make.data(dataset_path, chunksize = 1000)
# Fitting MLR Models
nmodel <- drglm::big.drglm(da,
formula = Var_1 ~ Var_2+ factor(Var_3)+factor(Var_4)+ factor(Var_5)+ Var_6,
10, family="gaussian")
# View the results table
print(nmodel)
# Fitting logistic Regression Model
bmodel <- drglm::big.drglm(da,
formula = factor(Var_3) ~ Var_1+ Var_2+ factor(Var_4)+ factor(Var_5)+ Var_6,
10, family="binomial")
# View the results table
print(bmodel)
# Fitting Poisson Regression Model
pmodel <- drglm::big.drglm(da,
formula = Var_5 ~ Var_1+ Var_2+ factor(Var_3)+ factor(Var_4)+ Var_6,
10, family="poisson")
# View the results table
print(pmodel)
```

`drglm` *Fitting Linear and Generalized Linear Model in "Divide and Recombine" approach to Large Data Sets*

Description

Function `drglm` aimed to fit GLMs to datasets larger in size that can be stored in memory. It uses popular divide and recombine technique to handle large data sets efficiently.Function `drglm` optimizes performance when linked with optimized BLAS libraries like ATLAS.The function `drglm` requires defining the number of chunks K and the fitfunction.The rest of the arguments are almost identical with the speedglm or biglm package.

Usage

```
drglm(formula, family, data, k, fitfunction)
```

Arguments

`formula`	An entity belonging to the "formula" class (or one that can be transformed into that class) represents a symbolic representation of the model that needs to be adjusted. Specifics about how the model is defined can be found in the 'Details' section.
`family`	An explanation of the error distribution that will be implemented in the model.
`data`	A data frame, list, or environment that is not required but can be provided if available.
`k`	Number of subsets to be used.
`fitfunction`	The function to be utilized for model fitting. `glm` or `speedglm` should be used.For Multinomial models, `multinom` function is preferred.

Value

A Generalized Linear Model is fitted in "Divide & Recombine" approach using "k" chunks to data set. A list of model coefficients is estimated using divide and recombine method with the respective standard error of estimates.

Author(s)

MH Nayem

References

- Xi, R., Lin, N., & Chen, Y. (2009). Compression and aggregation for logistic regression analysis in data cubes. IEEE Transactions on Knowledge and Data Engineering, 21(4).
- Chen, Y., Dong, G., Han, J., Pei, J., Wah, B. W., & Wang, J. (2006). Regression cubes with lossless compression and aggregation. IEEE Transactions on Knowledge and Data Engineering, 18(12).
- Zuo, W., & Li, Y. (2018). A New Stochastic Restricted Liu Estimator for the Logistic Regression Model. Open Journal of Statistics, 08(01).
- Karim, M. R., & Islam, M. A. (2019). Reliability and Survival Analysis. In Reliability and Survival Analysis.
- Enea, M. (2009) Fitting Linear Models and Generalized Linear Models with large data sets in R.
- Bates, D. (2009) Technical Report on Least Square Calculations.
- Lumley, T. (2009) *biglm* package documentation.

See Also

big.drglm, drglm.multinom

Examples

```
set.seed(123)
#Number of rows to be generated
n <- 10000
#creating dataset
dataset <- data.frame( pred_1 = round(rnorm(n, mean = 50, sd = 10)),
pred_2 = round(rnorm(n, mean = 7.5, sd = 2.1)),
pred_3 = as.factor(sample(c("0", "1"), n, replace = TRUE)),
pred_4 = as.factor(sample(c("0", "1", "2"), n, replace = TRUE)),
pred_5 = as.factor(sample(0:15, n, replace = TRUE)),
pred_6 = round(rnorm(n, mean = 60, sd = 5)))
#fitting MLRM
nmodel= drglm::drglm(pred_1 ~ pred_2+ pred_3+ pred_4+ pred_5+ pred_6,
data=dataset, family="gaussian", fitfunction="speedglm", k=10)
#Output
nmodel
#fitting simple logistic regression model
bmodel=drglm::drglm(pred_3~ pred_1+ pred_2+ pred_4+ pred_5+ pred_6,
```

```
data=dataset, family="binomial", fitfunction="speedglm", k=10)
#Output
bmodel
#fitting poisson regression model
pmodel=drglm::drglm(pred_5~ pred_1+ pred_2+ pred_3+ pred_4+ pred_6,
data=dataset, family="binomial", fitfunction="speedglm", k=10)
#Output
pmodel
#fitting multinomial logistic regression model
mmodel=drglm::drglm(pred_4~ pred_1+ pred_2+ pred_3+ pred_5+ pred_6,
data=dataset, family="multinomial", fitfunction="multinom", k=10)
#Output
mmodel
```

`drglm.multinom` *Fitting Multinomial Logistic Regression model in "Divide and Recombine" approach to Large Data Sets*

Description

Function `drglm.multinom` fits multinomial logistic regressiosn model to big data sets in divide and recombine approach.

Usage

```
drglm.multinom(formula, data, k)
```

Arguments

`formula`	An entity belonging to the "formula" class (or one that can be transformed into that class) represents a symbolic representation of the model that needs to be adjusted. Specifics about how the model is defined can be found in the 'Details' section.
`data`	A data frame, list, or environment that is not required but can be provided if available.
`k`	Number of subsets to be used.

Value

A "Multinomial (Polytomous) Logistic Regression Model" is fitted in "Divide and Recombine" approach.

Author(s)

MH Nayem

References

Karim, M. R., & Islam, M. A. (2019). Reliability and Survival Analysis. In Reliability and Survival Analysis. Venables WN, Ripley BD (2002). Modern Applied Statistics with S, Fourth edition. Springer, New York. ISBN 0-387-95457-0, https://www.stats.ox.ac.uk/pub/MASS4/.

See Also

big.drglm, drglm

Examples

```
set.seed(123)
#Number of rows to be generated
n <- 10000
#creating dataset
dataset <- data.frame( pred_1 = round(rnorm(n, mean = 50, sd = 10)),
pred_2 = round(rnorm(n, mean = 7.5, sd = 2.1)),
pred_3 = as.factor(sample(c("0", "1"), n, replace = TRUE)),
pred_4 = as.factor(sample(c("0", "1", "2"), n, replace = TRUE)),
pred_5 = as.factor(sample(0:15, n, replace = TRUE)),
pred_6 = round(rnorm(n, mean = 60, sd = 5)))
#fitting multinomial logistic regression model
mmodel=drglm::drglm.multinom(
pred_4~ pred_1+ pred_2+ pred_3+ pred_5+ pred_6, data=dataset, k=10)
#Output
mmodel
```

---

make.data *Reading Data File Larger than Memory for Fitting GLMs Using* big.drglm *Function*

Description

Reading Data File Larger than Memory for Fitting GLMs Using big.drglm Function

Usage

```
make.data(filename, chunksize, ...)
```

Arguments

filename	Path to the data set on disk.
chunksize	Size of the chunk or subset to be read from the large file for fitting GLMs.
...	Additional arguments to be passed to read.csv.

Value

A function that reads chunks of the data set.

Examples

```
# Create a toy dataset
set.seed(123)
# Number of rows to be generated
n <- 10000

# Creating dataset
dataset <- data.frame(
  Var_1 = round(rnorm(n, mean = 50, sd = 10)),
  Var_2 = round(rnorm(n, mean = 7.5, sd = 2.1)),
  Var_3 = as.factor(sample(c("0", "1"), n, replace = TRUE)),
  Var_4 = as.factor(sample(c("0", "1", "2"), n, replace = TRUE)),
  Var_5 = as.factor(sample(0:15, n, replace = TRUE)),
  Var_6 = round(rnorm(n, mean = 60, sd = 5))
)

# Save the dataset to a temporary file
temp_file <- tempfile(fileext = ".csv")
write.csv(dataset, file = temp_file, row.names = FALSE)

# Path to the temporary file
dataset_path <- temp_file
dataset_path  # Display the path to the temporary file

# Initialize the data reading function with the data set path and chunk size
da <- drglm::make.data(dataset_path, chunksize = 1000)

# Fitting MLR Models
nmodel <- drglm::big.drglm(da,
formula = Var_1 ~ Var_2 + factor(Var_3) + factor(Var_4) + factor(Var_5) + Var_6,
10, family = "gaussian")
# View the results table
print(nmodel)

# Fitting logistic Regression Model
bmodel <- drglm::big.drglm(da,
formula = factor(Var_3) ~ Var_1 + Var_2 + factor(Var_4) + factor(Var_5) + Var_6,
10, family = "binomial")
# View the results table
print(bmodel)

# Fitting Poisson Regression Model
pmodel <- drglm::big.drglm(da,
formula = Var_5 ~ Var_1 + Var_2 + factor(Var_3) + factor(Var_4) + Var_6,
10, family = "poisson")
# View the results table
print(pmodel)
```

Index