Package ‘elrm’

February 19, 2015

Type Package
Title Exact Logistic Regression via MCMC
Version 1.2.2
Date 2013-12-05
Depends R(>= 2.7.2), coda, graphics, stats
Author David Zamar, Jinko Graham, Brad McNeney
Maintainer David Zamar <zamar.david@gmail.com>
Description elrm implements a Markov Chain Monte Carlo algorithm to approximate exact conditional inference for logistic regression models. Exact conditional inference is based on the distribution of the sufficient statistics for the parameters of interest given the sufficient statistics for the remaining nuisance parameters. Using model formula notation, users specify a logistic model and model terms of interest for exact inference.
License GPL (>= 2)
URL http://stat-db.stat.sfu.ca:8080/statgen/research/elrm
Repository CRAN
Date/Publication 2013-12-07 00:52:08
NeedsCompilation yes

R topics documented:

  crashDat ................................................................. 2
diabDat ................................................................. 2
drugDat ................................................................. 3
elrm ................................................................. 3
plot.elrm ............................................................... 7
summary.elrm ......................................................... 8
titanDat ............................................................... 10
update.elrm ........................................................... 10
utiDat ................................................................. 13
### Crash Dataset: Calibration of Crash Dummies in Automobile Safety Tests

**Description**

Data from 58 simulated car crashes were analyzed. The relationship between the crash outcome (fatal, non-fatal) and 3 covariates was modeled.

**Usage**

```r
data(drugDat)
```

**Format**

Matrix with columns:

- **[,1]** `age` numeric: designed age of the crash dummy
- **[,2]** `vel` numeric: velocity of the car at impact
- **[,3]** `acl` numeric: acceleration of the car at impact
- **[,4]** `y` numeric: outcome from test (1=fatal, 0=non-fatal)
- **[,5]** `n` numeric: number of binomial trials. For this data, `n` is a vector of ones, thus each trial is modeled as a Bernoulli.

### Simulated Diabetes Dataset

**Description**

This dataset was simulated by sampling cases from an existing type 1 diabetes study (the original data could not be disclosed). The study investigates the relationship between concentration levels (low and high) of the islet antigen 2 antibody (IA2A) and several covariates of potential interest in type 1 diabetes patients (age, gender, and the number of copies (0,1 or 2) of the HLA-DQ2, HLA-DQ8 and HLA-DQ6.2 haplotypes).

**Usage**

```r
data(diabDat)
```

**Format**

Matrix with columns:

- **[,1]** `n` numeric: number of binomial trials.
- **[,2]** `IA2A` numeric: number of patients with high concentration levels of the islet antigen 2 antibody
elrm

\[ \begin{align*}
[.3] & \text{gender} & \text{numeric} & \text{gender of patient} \\
[.4] & \text{age} & \text{numeric} & \text{age of patient} \\
[.5] & \text{nDQ2} & \text{numeric} & \text{number of copies of the HLA-DQ2 haplotype (0, 1, or 2)} \\
[.6] & \text{nDQ8} & \text{numeric} & \text{number of copies of the HLA-DQ8 haplotype (0, 1, or 2)} \\
[.7] & \text{nDQ6.2} & \text{numeric} & \text{number of copies of the HLA-DQ6.2 haplotype (0, 1, or 2)}
\end{align*} \]

---

**drugDat**

*Drug Dataset*

**Description**

Simulated data for a hypothetical drug experiment comparing control versus treatment.

**Usage**

```r
data(drugDat)
```

**Format**

Matrix with columns:

- [.1] sex numeric gender (1=male, 0=female)
- [.2] treatment numeric treatment type (1=treatment, 0=control)
- [.3] recovered numeric number of subjects that recovered
- [.4] n numeric number of binomial trials

---

**elrm**

*elrm: exact-like inference in logistic regression models*

**Description**

elrm implements a modification of the Markov Chain Monte Carlo algorithm proposed by Forster et al. (2003) to approximate exact conditional inference for logistic regression models. The modifications can handle larger datasets than the original algorithm (Zamar 2006). Exact conditional inference is based on the distribution of the sufficient statistics for the parameters of interest given the sufficient statistics for the remaining nuisance parameters. Using model formula notation, users specify a logistic model and model terms of interest for exact inference.

**Usage**

```r
elrm(formula, interest, r = 4, iter = 1000, dataset, burnIn = 0, alpha = 0.05)
```
Arguments

- **formula**: a formula object that contains a symbolic description of the logistic regression model of interest in the usual R formula format. One exception is that the binomial response should be specified as `success/trials`, where `success` gives the number of successes and `trials` gives the number of binomial trials for each row of `dataset`.

- **interest**: a formula object that contains a symbolic description of the model terms for which exact conditional inference is of interest.

- **r**: a parameter of the MCMC algorithm that influences how the Markov chain moves around the state space. Small values of `r` cause the chain to take small, relatively frequent steps through the state space; larger values cause larger, less frequent steps. The value of `r` must be an even integer less than or equal to the length of the response vector. Typical values are 4, 6 or 8; default=4.

- **iter**: an integer representing the number of Markov chain iterations to make (must be larger than or equal to 1000); default=1000.

- **dataset**: a data.frame object where the data are stored.

- **burnIn**: the burn-in period to use when conducting inference. Values of the Markov chain in the burn-in period are discarded; default=0.

- **alpha**: determines the level used for confidence intervals; default=0.05.

Details

The function `summary()` (i.e., `summary.elrm`) can be used to obtain or print a summary of the results.

Each estimated exact p-value is based on the conditional probabilities test.

The Monte Carlo standard error of each p-value is computed by the batch-means method (Geyer C.J. 1992).

Inference on each parameter must be based on a Markov chain of at least 1000 iterations, otherwise NA is returned.

If the observed value of the sufficient statistic for a parameter is either the maximum or the minimum value sampled, the MUE of the parameter is given instead of the CMLE. In such cases, the resulting confidence interval is open-ended on one side.

Apart from the documentation files accompanying this package, the `elrm` package vignette may be downloaded from the `elrm` package homepage on the SFU Statistical Genetics Working Group website (http://stat-db.stat.sfu.ca:8080/statgen/research/elrm). The vignette is also distributed with the code.

Value

- **coeffs**: a vector containing the parameter estimates.
coefs.ci a list containing (1-\alpha)*100% confidence intervals for each parameter of interest.

distribution a list containing distribution tables for the sampled values of the sufficient statistic of the parameters of interest conditional on all the rest.

mc an mcmc object containing the Markov chain of sampled values of the sufficient statistics for the parameters of interest. Columns correspond to parameters; rows to Monte Carlo iterations.

obs.suff.stat a vector containing the observed value of the sufficient statistic for each parameter of interest.

call.history a list composed of the matched call and the history of calls to update().

data set the data.frame object that was passed to elrm() as an argument.

last the last response vector sampled by the Markov chain.

mc.size a vector containing the lengths of the extracted Markov chains used in testing each parameter. The length of the Markov chain used for the joint test (i.e., iter) is also included as the first element.

ci.level the level used when constructing the confidence intervals for the parameters of interest. The level is calculated as (1-\alpha)*100%.

p.values a vector containing the estimated p-value for jointly testing that the parameters of interest are simultaneously equal to zero, and the full conditional p-values from separately testing each parameter equal to zero.

p.values.se a vector containing the Monte Carlo standard errors of the estimated p-values of each term of interest.

mc.size a vector containing the Monte Carlo standard errors of the estimated p-values of each term of interest.

Warning

The labels of the terms in the in the interest model should match those found in the formula model. Thus, the term.labels attribute of terms.formula(interest) should match those found in terms.formula(formula). Please see the Examples section for more details.

Author(s)

David Zamar, Jinko Graham, Brad McNeney

References


See Also
update.elrm, summary.elrm, plot.elrm.

Examples

# Drug dataset example with sex and treatment as the variables of interest
data(drugdat);
drug.elrm = elrm(formula=recovered/n~sex+treatment, interest=-sex+treatment, r=4, iter=50000, burnIn=1000, dataset=drugDat);

## Not run:
# crash dataset example where the terms of interest are age and
# the interaction of age and velocity.
data(crashDat);
# The following call produces the error message shown below.
crash.elrm = elrm(formula=y/n~vel+age+acl+age:vel, interest=-age+age:vel, r=4, iter=5000, dataset=crash, burnIn=100);

# Error in getDesignMatrix(formula, interest, dataset = dataset) :
# the 'term.labels' attribute of 'terms.formula(interest)' must match those
# found in 'terms.formula(formula)'

# The error occurs, because the variables within the interaction term in a formula
# are re-ordered by the ordering in which the variables occur. Thus, the interaction
# between age and velocity is labeled as vel:age in the 'formula' model and as age:vel
# in the 'interest' model.

attr(terms.formula(y/n~vel+age+acl+age:vel),"term.labels");
# [1] "vel"  "age"  "acl"  "vel:age"

attr(terms.formula(~age+age:vel),"term.labels");
# [1] "age"  "age:vel"

# To get around this problem, place age before vel in the 'formula' model.
crash.elrm = elrm(formula=y/n~age+vel+acl+age:vel, interest=-age+age:vel, r=4, iter=5000, dataset=crash, burnIn=100);

## End(Not run)

## Not run:
# Urinary tract dataset example with dia as the variable of interest
data(utiDat);
uti.elrm = elrm(uti/n~age+current+dia+oc+pastyr+vi+vic+vicl+vis, interest=-dia, r=4, iter=30000,burnIn=1000, dataset=utiDat);
## End(Not run)

## Not run:
# Titanic dataset example where the variable of interest, class, is coded as a factor
data(titanDat);
titanic.elrm = elrm(surv/n~as.factor(class)+age+sex, interest=-as.factor(class), r=4, iter=50000, burnIn=1000, dataset=titanDat);
## End(Not run)
plot.elrm

Plot Diagnostics for an elrm Object

Description

Produces both a trace plot and histogram of the sampled values of each sufficient statistic of interest. Sampled values within the burn-in period are also plotted.

Usage

## S3 method for class 'elrm'
plot(x, p = 1, breaks = "Sturges", ask=FALSE, ...)

Arguments

- **x**: an object of class elrm, resulting from a call to elrm() or a previous call to update().
- **p**: the sampling fraction of points to be plotted. A random sample consisting of \( p \times 100\% \) of all the observations in the Markov chain is plotted; default=1.
- **breaks**: a vector giving the number of cells to use for the histogram of each sufficient statistic of interest or a single number giving the number of cells for each histogram or the character string naming an algorithm to compute the number of cells.
- **ask**: the graphics parameter ask: see par for details. If set to TRUE will ask between plots corresponding to each sufficient statistic; default=FALSE.
- **...**: additional arguments to the plot function (currently unused).

Details

The default for breaks is "Sturges": see nclass.Sturges. Other names for which algorithms are supplied are "Scott" and "FD".

Author(s)

David Zamar, Jinko Graham, Brad McNeney

References


See Also

update.elrm, summary.elrm, elrm.
Examples

# Drug dataset example with sex and treatment as the variables of interest
data(drugDat);
drug.elrm = elrm(formula=recovered/n~sex+treatment, interest=~sex+treatment, r=4,
iter=50000, burnIn=1000, dataset=drugDat);

# Plot the sampled values of the sufficient statistic for each parameter of
# interest (sex and treatment)
plot(drug.elrm,p=0.5,ask=TRUE);

summary.elrm

Summarize an elrm Object

Description

Summary method for class elrm that formats and prints out the results of an elrm object.

Usage

## S3 method for class 'elrm'
summary(object, ...)

Arguments

object an object of class elrm, resulting from a call to elrm() or a previous call to
update().
...
additional arguments to the summary function (currently unused).

Details

The following results are formatted and printed to the screen: the matched call, coefficient estimates
and confidence intervals for each model term of interest, estimated p-value for jointly testing that
the parameters of interest are simultaneously equal to zero, full conditional p-values from separately
testing each parameter equal to zero, length of the Markov chain that inference was based on, and
the Monte Carlo standard error of each reported p-value.

Author(s)

David Zamar, Jinko Graham, Brad McNeney

References

Zamar David. Monte Carlo Markov Chain Exact Inference for Binomial Regression Models. Mas-
Zamar D, McNeney B and Graham J. elrm: Software Implementing Exact-like Inference for Logistic
See Also

update.elrm, plot.elrm, elrm.

Examples

```r
# Drug dataset example with both sex and treatment as the variables of interest
data(drugdat)
drug.elrm = elrm(formula=recovered/n~sex+treatment, interest=-sex+treatment, r=4,
iter=50000, burnIn=1000, dataset=drugDat);

# Summarize the results:
summary(drug.elrm);

# Call:
# [[1]]
# elrm(formula = recovered/n ~ sex + treatment, interest = ~sex +
# treatment, r = 4, iter = 1e+05, dataset = drugDat, burnIn = 1000)

# Results:
# estimate p-value p-value_se mc_size
# joint NA 0.12951 0.00216 99000
# sex 0.29479 0.54092 0.00880 2749
# treatment 0.82389 0.06892 0.00347 13131

# 95% Confidence Intervals for Parameters

# lower upper
# sex -0.6109481 1.209525
# treatment -0.1042183 2.028083

# Urinary tract dataset example with dia as the variable of interest
data(utiDat)
uti.elrm = elrm(uti/n~age+current+dia+oc+pastyr+vi+vicl+vis, interest=~dia, r=4,
iter=30000, burnIn=100, dataset=utiDat);

# Summarize the results:
summary(uti.elrm);

# Call:
# [[1]]
# elrm(formula = uti/n ~ age + current + dia + oc + pastyr + vi +
# vic + vicl + vis, interest = ~dia, r = 4, iter = 30000, dataset = uti,
# burnin = 1000)

# Results:
# estimate p-value p-value_se mc_size
# dia 2.07146 0.03286 0.00802 29000

# 95% Confidence Intervals for Parameters
```
titanDat  Titanic Dataset

Description
Relationship between survival and passenger class on the Titanic. The records of the sinking of the
Titanic were studied to establish the relationship between survival and passenger class on the ship.
For each person on board the ocean liner, this dataset records sex, age (child/adult), class (crew, 1st,
2nd, 3rd class) and whether or not the person survived.

Usage
data(titanDat)

Format
Matrix with columns:

<table>
<thead>
<tr>
<th>Column 1</th>
<th>Column 2</th>
<th>Column 3</th>
<th>Column 4</th>
<th>Column 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>surv</td>
<td>n</td>
<td>class</td>
<td>age</td>
<td>sex</td>
</tr>
<tr>
<td>numeric</td>
<td>numeric</td>
<td>numeric</td>
<td>numeric</td>
<td>numeric</td>
</tr>
<tr>
<td>number of survivors</td>
<td>total number of people</td>
<td>passenger class (0 = crew, 1 = first, 2 = second, 3 = third)</td>
<td>age group (1 = adult, 0 = child)</td>
<td>gender (1 = male, 0 = female)</td>
</tr>
</tbody>
</table>

References

update.elrm  Update Method for Objects of Class elrm.

Description
An update method for objects created by elrm(). Extends the Markov chain of an elrm object by a
specified number of iterations.

Usage
## S3 method for class 'elrm'
update(object, iter, burnin = 0, alpha = 0.05, ...)

# lower upper
# dia -0.06231932 Inf

## End(Not run)
Arguments

- **object**: an object of class `elrm`, resulting from a call to `elrm()` or a previous call to `update()`.
- **iter**: an integer representing the number of Markov chain iterations to make.
- **burnIn**: the burn-in period to use when conducting inference. Values of the Markov chain in the burn-in period are discarded; default=0.
- **alpha**: determines the level used for confidence intervals; default=0.05.
- **...**: additional arguments to the update function (currently unused).

Details

Extends the Markov chain of an `elrm` object by creating a new Markov chain of the specified length using the last sampled value as the starting point. The newly created chain is then appended to the original. Subsequent inference is based on the extended Markov chain.

Value

An object of class `elrm`.

Author(s)

David Zamar, Jinko Graham, Brad McNeney

References


See Also

`summary.elrm`, `plot.elrm`, `elrm`.

Examples

# Drug dataset example with sex and treatment as the variables of interest
data(drugDat);
drug.elrm = elrm(formula=recovered/n ~ sex + treatment, interest=-sex + treatment, r=4, iter=2000, burnIn=0, dataset=drugDat);

# Summarize the results
summary(drug.elrm);

# Call:
# [[1]]
# elrm(formula = recovered/n ~ sex + treatment, interest = ~sex +
#     treatment, r = 4, iter = 2000, dataset = drugDat, burnIn = 0)
update.elrm

# Results:
# estimate p-value p-value_se mc_size
# joint NA 0.517 0.01755 2000
# sex NA NA NA 90
# treatment NA NA NA 275

# 95% Confidence Intervals for Parameters
# lower upper
# sex NA NA
# treatment NA NA

# Call update and extend the chain by 50000 iterations and set the burn-in
# period to 100 iterations
drug.elrm = update(drug.elrm, iter=50000, burnIn=100);

# Summarize the results
summary(drug.elrm);

# Call:
# [[1]]
# elrm(formula = recovered/n ~ sex + treatment, interest = ~sex +
# treatment, r = 4, iter = 2000, dataset = drugDat, burnIn = 0)
# [[2]]
# update.elrm(object = drug.elrm, iter = 50000, burnIn = 100)

# Results:
# estimate p-value p-value_se mc_size
# joint NA 0.14669 0.00314 51900
# sex 0.29431 0.52625 0.01100 1543
# treatment 0.75707 0.07805 0.00512 6842

# 95% Confidence Intervals for Parameters
# lower upper
# sex -0.6109599 1.230676
# treatment -0.1366174 1.845202

# Now change the burn-in to 5000
drug.elrm = update(drug.elrm, iter=0, burnIn=5000);

# Summarize the results
summary(drug.elrm);

# Call:
# [[1]]
# elrm(formula = recovered/n ~ sex + treatment, interest = ~sex +
# treatment, r = 4, iter = 2000, dataset = drugDat, burnIn = 0)
# [[2]]
# update.elrm(object = drug.elrm, iter = 50000, burnIn = 100)

# Results:
# estimate     p-value  p-value_se  mc_size
# joint          NA    0.13419     0.00341 47000
# sex            0.28423 0.52774     0.01890 1370
# treatment      0.79565 0.07500     0.00377 6227

# 95% Confidence Intervals for Parameters
# lower    upper
# sex     -0.0053313 1.199807
# treatment -0.1240906 1.926238

---

**utiDat**

**Urinary Tract Infection and Contraceptive Use**

**Description**

How is the development of first-time urinary tract infection (UTI) related to contraceptive use? A study of sexually active college women with UTI was conducted, and their use of various contraceptives was surveyed.

**Usage**

data(utiDat)

**Format**

Matrix with columns:

- [.,1] uti numeric infection status (1=yes, 0=no)
- [.,2] n numeric number of binomial trials
- [.,3] age numeric age of the person
- [.,4] current numeric no regular current sex partner (1=yes, 0=no)
- [.,5] dia numeric diaphragm use (1=yes, 0=no)
- [.,6] oc numeric oral contraceptive (1=yes, 0=no)
- [.,7] pastyr numeric no regular partner with relationship < 1yr (1=yes, 0=no)
- [.,8] vi numeric vaginal intercourse (1=yes, 0=no)
- [.,9] vic numeric vaginal intercourse with condom (1=yes, 0=no)
- [.,10] vicl numeric vaginal intercourse with lubricated condom (1=yes, 0=no)
- [.,11] vis numeric vaginal intercourse with spermicide (1=yes, 0=no)
References

Index

*Topic datasets
  crashDat, 2
  diabDat, 2
  drugDat, 3
  titanDat, 10
  utiDat, 13

*Topic methods
  elrm, 3
  plot.elrm, 7
  summary.elrm, 8
  update.elrm, 10

  crashDat, 2
  diabDat, 2
  drugDat, 3
  elrm, 3, 7, 9, 11
  mcmc, 5
  nclass.Sturges, 7
  par, 7
  plot.elrm, 6, 7, 9, 11
  summary.elrm, 4, 6, 7, 8, 11
  titanDat, 10
  update.elrm, 6, 7, 9, 10
  utiDat, 13