Package 'BayesLogit'

May 6, 2012

Version 0.1-0

Date 2012-05-06

Depends R (>= 2.14.0)

Title Logistic Regression

Author Nicolas Polson, James G. Scott, and Jesse Windle

Maintainer Jesse Windle <jwindle@ices.utexas.edu>

Description The BayesLogit package does posterior simulation for binary and multinomial logistic regression using the Polya-Gamma latent variable technique. This method is fully automatic, exact, and fast. A routine to efficiently sample from the Polya-Gamma class of distributions is included.

License GPL (>= 3)

Repository CRAN

Date/Publication 2012-05-06 14:11:24

R topics documented:

`logit` *Default Bayesian Logistic Regression*

Description

Run a Bayesian logistic regression.

Usage

```
logit(y, X, n=rep(1,length(y)),
      y.prior=0.5, x.prior=colMeans(as.matrix(X)), n.prior=1.0,
      samp=1000, burn=500)
```

Arguments

`y`	An N dimensional vector; y_i is the average response at x_i.
`X`	An N x P dimensional design matrix; x_i is the ith row.
`n`	An N dimensional vector; n_i is the number of observations at each x_i.
`y.prior`	Average response at `x.prior`.
`x.prior`	Prior predictor variable.
`n.prior`	Number of observations at `x.prior`.
`samp`	The number of MCMC iterations saved.
`burn`	The number of MCMC iterations discarded.

Details

Logistic regression is a classification mechanism. Given the binary data $\{y_i\}$ and the p-dimensional predictor variables $\{x_i\}$, one wants to forecast whether a future data point y* observed at the predictor x* will be zero or one. Logistic regression stipulates that the statistical model for observing a success=1 or failure=0 is governed by

$$P(y^* = 1|x^*, \beta) = (1 + \exp(-x * \beta))^{-1}.$$

Instead of representing data as a collection of binary outcomes, one may record the average response y_i at each unique x_i given a total number of n_i observations at x_i. We follow this method of encoding data.

Polson and Scott suggest placing a Jeffrey's Beta prior Be(1/2,1/2) on

$$m(\beta) := P(y_0 = 1|x_0, \beta) = (1 + \exp(-x_0\beta))^{-1},$$

which generates a Z-distribution prior for β,

$$p(\beta) = \exp(0.5x_0\beta)/(1 + \exp(0.5x_0\beta)).$$

One may interpret this as "prior" data where the average response at x_0 is $1/2$ based upon a "single" observation. The default value of $x_0 = mean(x), x = \{x_i\}$.

Value

`logit` returns a list.

`beta`	A samp x P array; the posterior sample of the regression coefficients.
`w`	A samp x N' array; the posterior sample of the latent variable. WARNING: N' may be less than N if data is combined.
`y`	The response matrix–different than input if data is combined.
`X`	The design matrix–different than input if data is combined.
`n`	The number of samples at each observation–different than input if data is combined.

References

Nicolas G. Polson, James G. Scott, and Jesse Windle. Bayesian inference for logistic models using Polya-Gamma latent variables. http://arxiv.org/abs/1205.0310

Nicolas Poslon and James G. Scott. Default Bayesian analysis for multi-way tables: a data-augmentation approach. http://arxiv.org/pdf/1109.4180

See Also

rpg, logit.EM, mlogit

Examples

```
## From UCI Machine Learning Repository.
data(spambase);

## A subset of the data.
sbase = spambase[seq(1,nrow(spambase),10),];

X = model.matrix(is.spam ~ word.freq.free + word.freq.1999, data=sbase);
y = sbase$is.spam;

## Run logistic regression.
output = logit(y, X, samp=1000, burn=100);
```

---

logit.EM *Logistic Regression Expectation Maximization*

Description

Expectation maximization for logistic regression.

Usage

```
logit.EM(y, X, n=rep(1,length(y)),
            y.prior=0.5, x.prior=colMeans(as.matrix(X)), n.prior=1.0,
            tol=1e-9, max.iter=100)
```

Arguments

y	An N dimensional vector; y_i is the average response at x_i.
X	An N x P dimensional design matrix; x_i is the ith row.
n	An N dimensional vector; n_i is the number of observations at each x_i.
y.prior	Average response at x.prior.
x.prior	Prior predictor variable.
n.prior	Number of observations at x.prior.
tol	Threshold at which algorithm stops.
max.iter	Maximum number of iterations.

Details

Logistic regression is a classification mechanism. Given the binary data $\{y_i\}$ and the p-dimensional predictor variables $\{x_i\}$, one wants to forecast whether a future data point y* observed at the predictor x* will be zero or one. Logistic regression stipulates that the statistical model for observing a success=1 or failure=0 is governed by

$$P(y^* = 1|x^*, \beta) = (1 + \exp(-x * \beta))^{-1}.$$

Instead of representing data as a collection of binary outcomes, one may record the average response y_i at each unique x_i given a total number of n_i observations at x_i. We follow this method of encoding data.

Polson and Scott suggest placing a Jeffrey's Beta prior Be(1/2,1/2) on

$$m(\beta) := P(y_0 = 1|x_0, \beta) = (1 + \exp(-x_0\beta))^{-1},$$

which generates a Z-distribution prior for β,

$$p(\beta) = \exp(0.5x_0\beta)/(1 + \exp(0.5x_0\beta)).$$

One may interpret this as "prior" data where the average response at x_0 is $1/2$ based upon a "single" observation. The default value of $x_0 = mean(x), x = \{x_i\}$.

Value

beta	The posterior mode.
iter	The number of iterations.

References

Nicolas G. Polson, James G. Scott, and Jesse Windle. Bayesian inference for logistic models using Polya-Gamma latent variables. http://arxiv.org/abs/1205.0310

Nicolas Poslon and James G. Scott. Default Bayesian analysis for multi-way tables: a data-augmentation approach. http://arxiv.org/pdf/1109.4180

See Also

rpg, logit, mlogit

Examples

```
## From UCI Machine Learning Repository.
data(spambase);

## A subset of the data.
sbase = spambase[seq(1,nrow(spambase),10),];

X = model.matrix(is.spam ~ word.freq.free + word.freq.1999, data=sbase);
y = sbase$is.spam;

## Run logistic regression.
output = logit.EM(y, X);
```

---

mlogit *Bayesian Multinomial Logistic Regression*

Description

Run a Bayesian multinomial logistic regression.

Usage

```
mlogit(y, X, n=rep(1,nrow(as.matrix(y))),
       m.0=array(0, dim=c(ncol(X), ncol(y))),
       P.0=array(0, dim=c(ncol(X), ncol(X), ncol(y))),
       samp=1000, burn=500)
```

Arguments

y	An N x J-1 dimensional matrix; y_{ij} is the average response for category j at x_i.
X	An N x P dimensional design matrix; x_i is the ith row.
n	An N dimensional vector; n_i is the total number of observations at each x_i.

m.0	A P x J-1 matrix with the $beta_j$'s prior means.
P.0	A P x P x J-1 array of matrices with the $beta_j$'s prior precisions.
samp	The number of MCMC iterations saved.
burn	The number of MCMC iterations discarded.

Details

Multinomial logistic regression is a classifiction mechanism. Given the multinomial data $\{y_i\}$ with J categories and the p-dimensional predictor variables $\{x_i\}$, one wants to forecast whether a future data point y* at the predictor x*. Multinomial Logistic regression stiuplates that the statistical model for observing a draw category j after rolling the multinomial die $n^* = 1$ time is governed by

$$P(y^* = j|x^*, \beta, n^* = 1) = e^{x^*\beta_j} / \sum_{k=1}^{J} e^{x^*\beta_k}.$$

Instead of representing data as the total number of responses in each category, one may record the average number of responses in each category and the total number of responses n_i at x_i. We follow this method of encoding data.

We assume that $\beta_J = 0$ for purposes of identification!

You may use mlogit for binary logistic regression with a normal prior.

Value

mlogit returns a list.

beta	A samp x P x J-1 array; the posterior sample of the regression coefficients.
w	A samp x N' x J-1 array; the posterior sample of the latent variable. WARNING: N' may be less than N if data is combined.
y	The response matrix–different than input if data is combined.
X	The design matrix–different than input if data is combined.
n	The number of samples at each observation–different than input if data is combined.

References

Nicolas G. Polson, James G. Scott, and Jesse Windle. Bayesian inference for logistic models using Polya-Gamma latent variables. http://arxiv.org/abs/1205.0310

See Also

rpg, logit.EM, logit

Examples

```
## Use the iris dataset.
N = nrow(iris)
P = ncol(iris)
J = nlevels(iris$Species)

X     = model.matrix(Species ~ ., data=iris);
y.all = model.matrix(~ Species - 1, data=iris);
y     = y.all[,-J];

out = mlogit(y, X, samp=1000, burn=100);
```

`rks` *The Kolmogorov-Smirnov distribution*

Description

Generate a random variate from the Kolmogorov-Smirnov distribution.

This is not directly related to the Polya-Gamma technique, but it is a nice example of using an alternating sum to generate a random variate.

Usage

```
rks(N=1)
```

Arguments

`N`	The number of random variates to generate.

Details

The density function of the KS distribution is

$$f(x) = 8 \sum_{i=1}^{\infty} (-1)^{n+1} n^2 x e^{-2n^2 x^2}.$$

We follow Devroye (1986) p. 161 to generate random draws from KS.

References

L. Devroye. Non-Uniform Random Variate Generation, 1986.

Examples

```
X = rks(1000)
```

---

rpg *The Polya-Gamma Distribution*

Description

Generate a random variate from the Polya-Gamma distribution.

Usage

```
rpg.gamma(num=1, n=1, z=0.0, trunc=200)

rpg.devroye(num=1, n=1, z=0.0)
```

Arguments

	You may call rpg when n and z are vectors.
	The number of random variates to simulate.
num	Degrees of freedom.
z	Parameter associated with tilting.
trunc	The number of elements used in sum of gammas approximation.

Details

A random variable X with distribution PG(n,z) is generated by

$$X \sim 2.0 \sum k = 1^{\infty} G(n,1)/((k-1/2)^2 4.0\pi^2 + z^2).$$

The density for X may be derived from Z and PG(n,0) as

$$p(x|n,z) \propto \exp(-z^2/2x) p(x|n,0).$$

Thus PG(n,z) is an exponentially tilted PG(n,0).

Two different methods for generating this random variable are implemented. In general, you may use rpg.gamma to generate an approximation of PG(n,z) using the sum of Gammas representation above. When n is a natural number you may use `rpg.devroye` to sample PG(n,z). The later method is fast.

Value

This function returns num Polya-Gamma samples.

References

Nicolas G. Polson, James G. Scott, and Jesse Windle. Bayesian inference for logistic models using Polya-Gamma latent variables. http://arxiv.org/abs/1205.0310

See Also

logit.EM, logit, mlogit

Examples

```
a = c(1, 2, 3);
b = c(4, 5, 6);

## If a is only integers, use Devroye-like method.
X = rpg.devroye(100, a, b);

a = c(1.2, 2.3, 3.2);
b = c(4, 5, 6);

## If a has scalars use sum-of-gammas method.
X = rpg.gamma(100, a, b);
```

---

spambase *Spambase Data*

Description

The spambase data has 57 real valued explanatory variables which characterize the contents of an email and and one binary response variable indicating if the email is spam. There are 4601 observations.

Format

A data frame: the first column is a binary response variable indicating if the email is spam. The remaining 57 columns are real valued explanatory variables.

Details

Of the 57 explanatory variables, 48 describe word frequency, 6 describe character frequency, and 3 describe sequences of capital letters.

word.freq.<word> A continuous explanatory variable describing the frequency with which the word <word> appears; measured in percent.

char.freq.<char> A continuous explanatory variable describing the frequency with which the character <char> appears; measured in percent.

capital.run.length.<oper> A statistic involving the length of consecutive capital letters.

Use `names` to see the specific words, characters, or statistics for each respective class of variable.

References

Mark Hopkins, Erik Reeber, George Forman, and Jaap Suermondt of Hewlett-Packard Labs (1999). Spambase Data Set. http://archive.ics.uci.edu/ml/datasets/Spambase

Frank, A. & Asuncion, A. (2010). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

Index