Package 'postMCMCglmm'

April 17, 2013

Type Package

Version 0.1-2

Date 2013-4-17

Title Average marginal predicted probabilities from Bayesian ordered probit models

Author Joshua Wiley <jwiley.psych@gmail.com>

Maintainer Joshua Wiley <jwiley.psych@gmail.com>

Depends R (>= 2.15.2), methods, stats, MCMCglmm (>= 2.17), coda (>= 0.16-1)

Imports ggplot2 (>= 0.9.3), grid, nlme, emdbook

Description R package to support Modern Modeling Methods presentation on using
average marginal predicted probabilities as an intuitive way to present
results from a Bayesian cross classified hierarchical ordered probit model.

License GPL-3

KeepSource true

ByteCompile true

URL http://github.com/jwiley/postMCMCglmm/

Collate 'extraction.R' 'postMCMCglmm.R' 'misc.R' 'prediction.R'

R topics documented:

fixef.MCMCglmm   *Extract fixed effects from an* MCMCglmm *object*

Description

Function designed to extract the fixed effects from an MCMCglmm model object. Can either extract all samples from the fixed effects posteriors or return the posterior means.

Usage

```
 ## S3 method for class 'MCMCglmm'
fixef(object, use = c("all", "mean"),
   ...)
```

Arguments

object	An MCMCglmm model object to extract the effects from
use	A character string indicating whether to extract all posterior samples or the mean of the posteriors. Defaults to "all".
...	Arguments passed on to the worker function.

Value

A matrix of the fixed effects

See Also

ranef.MCMCglmm

Examples

```
## Not run:
  # a simple MCMCglmm model
  data(PlodiaPO)
  m <- MCMCglmm(PO ~ 1, random= ~ FSfamily, data=PlodiaPO, verbose=FALSE)

  # only extract average fixed effects
  fixef(m, use = "mean")

  # histogram of posterior samples of fixed effects
  hist(fixef(m))
  # matches the mean
  rowMeans(fixef(m))

## End(Not run)
```

---

jointPosterior   *Plot the joint posterior from an MCMC*

Description

A function to calculate the HPD region for a joint posterior, and plot the bivariate density with HPD region, and univariate densities on the margins using the **ggplot2** package.

Usage

```
jointPosterior(dat, x, y, xlab = "", ylab = "",
  probs = 0.95, plot = TRUE, topleftmargin = 0.2)
```

Arguments

dat	A data frame, typically of MCMC posterior samples.
x	A character string of the variable in dat to plot on the x-axis.
y	A character string of the variable in dat to plot on the y-axis.
xlab	A character string or expression containing the x-axis label. Expressions may be used for Greek or mathematical symbols. Defaults to blank.
ylab	A character string or expression containing the x-axis label. Expressions may be used for Greek or mathematical symbols. Defaults to blank.
probs	A numeric value of the probability to use for calculating the HPD region for the two dimensional density plot. Passed to HPDregionplot from the **emdbook** package.
plot	A logical whether the graph should be plotted or not. Defaults to TRUE.
topleftmargin	A numeric value of the left margin of the upper univariate density plot. Need to adjust this to get the upper univariate density to line up with the bivariate density plot.

Value

An invisible list of the univariate and bivariate density plots and the viewport used to put them all together. Primarily called for the side effect of creating a graph.

Examples

```
## Not run:
  # sample data
  set.seed(10)
  dens2dtestdat <- as.data.frame(MASS::mvrnorm(4500, c(b1 = -.1, b2 = .05),
    Sigma = c(.05, .02)*matrix(c(1, -.5, -.5, 1), 2)*rep(c(.05, .02), each = 2)))
  d <- as.data.frame(mar2c$Sol[, 10:11]); colnames(d) <- c("b1", "b2")
  tmp <- as.data.frame(HPDregionplot(as.mcmc(d), n = 200)[[1]])
  jointPosterior(d, x = "b1", y = "b2", tmp, xlab = "Reactivity x Support", ylab = "Recovery x Support")

  # make the plot
  jointPosterior(dens2dtestdat, x = "b1", y = "b2", xlab = "Time x Constraint",
    ylab = bquote(Time^2 ~ x ~ Constraint))
  # clean up
```

```
  rm(dens2dtestdat)

## End(Not run)
```

---

paramNamesMCMCglmm   *Extract the parameter names from an* MCMCglmm *object*

Description

Simple function to extract the fixed and random effects parameter names from an MCMCglmm object.

Usage

```
paramNamesMCMCglmm(object, ...)
```

Arguments

object   An MCMCglmm object

...   not used

Value

A list with two elements:

fixed   A character vector of the fixed effects parameter names

random   A character vector of the random effects parameter names

See Also

[fixef.MCMCglmm](), [ranef.MCMCglmm]()

Examples

```
## Not run:
  # a simple MCMCglmm model
  data(PlodiaPO)
  m <- MCMCglmm(PO ~ 1, random = ~ FSfamily, data = PlodiaPO, verbose=FALSE)

  # extract the parameter names
  paramNamesMCMCglmm(m)

## End(Not run)
```

---

postMCMCglmm   *Post Estimation for MCMCglmm*

Description

Post Estimation for MCMCglmm

Details

postMCMCglmm is a simple package to support a proposal at Modern Modeling Methods 2013 It has extraction functions and prediction functions for MCMCglmm models to help get average marginal predicted probabilities from a probit model.

Package:	postMCMCglmm
Type:	Package
Version:	0.1-1
Date:	2013-1-30
License:	GPL (>= 3)
LazyLoad:	yes

Author(s)

Joshua Wiley <jwiley.psych@gmail.com>,

Maintainer: Joshua Wiley <jwiley.psych@gmail.com>

See Also

See `fixef.MCMCglmm`,

Examples

```
## Not run:
  # fill me in!!

## End(Not run)
```

---

predict2   *Define a generic prediction function*

Description

This defintes a generic `predict2` function which is similar to the usual `predict` but can use different methods. In particular, the `MCMCglmm` method has features not available in the regular `predict` method for `MCMCglmm` objects.

This is the main workhorse of the package. It is a `predict2` method for MCMCglmm objects. There are a few core arguments. The model `object` and design matrices, X (fixed effects) and Z (random effects). If X and Z are missing, it will attempt to fill them in from the model object (which optionally saves them). If X and Z are specified or `NULL`, they are not used. This is useful either

for out of sample predictions or to use just the fixed effects. Note that these must be full design matrices, not data matrices. For example, they must dummy code factors and include the intercept (if there was an intercept in the model).

Usage

```
predict2(object, ...)

## S3 method for class 'MCMCglmm'
predict2(object, X, Z,
   use = c("all", "mean"), type = c("lp", "response"),
   ...)
```

Arguments

`object`	A model object to predict from
`...`	Additional arguments passed to the methods
`X`	The fixed effects design matrix. Can be the original or new data.
`Z`	The random effects design matrix. Can be the original or new data.
`use`	A character string. Use just the posterior "mean" or "all" posterior samples (the default).
`type`	A cahracter string. Either "lp" for the linear predictor (the default) or "response" for the predicted values on the response scale.

Details

You can also use all posterior samples or just the mean. All is nice because it lets you construct highest posterior density (HPD) intervals around the predicted values, rather than just get an estimate. The mean is nice because if that is all you care about, it is much much faster. You can get either the linear predictor values or the response scale. However, response is currently only implemented for ordinal (probit) models. Theoretically it could be extended but the code is a pain.

Value

Either a matrix of the linear predictor if type =   "lp" or a list of class MCMCglmmPredictedProbs if type = "response"

See Also

[summary.MCMCglmmPredictedProbs](), [recycler]()

Examples

```
# to see available methods
methods(predict2)
## Not run:
   data(PlodiaPO)
   PlodiaPO <- within(PlodiaPO, {
     PO2 <- cut(PO, quantile(PO, c(0, .33, .66, 1)))
   })

   m <- MCMCglmm(PO2 ~ 1, random = ~ FSfamily,
     family = "ordinal", data = PlodiaPO,
```

```
      prior = list(
        R = list(V = 1, fix = 1),
        G = list(
          G1 = list(V = 1, nu = .002)
        )
      ), verbose=FALSE, thin=1, pr=TRUE)

  # predicted probabilities for each level of the outcome
  # using all posterior samples
  yhat <- predict2(m, use = "all", type = "response")
  str(yhat) # view structure

  # summary of predicted probabilities
  sumyhat <- summary(yhat)
  # first few summaries for level 1
  head(sumyhat[[1]])

  # first few summaries for level 2
  head(sumyhat[[2]])

  # first few summaries for level 3
  head(sumyhat[[3]])

  # combine
  longsum <- do.call(rbind.data.frame, sumyhat)
  # create a level indicator
  longsum$Level <- factor(rep(1:3, each = nrow(sumyhat[[1]])))

  # plot
  boxplot(M ~ Level, data = longsum)

## End(Not run)
```

---

print.postMCMCglmmRE	*Print method for extracted random effects (as standard deviations) from a MCMCglmm object*

Description

Print method for extracted random effects (as standard deviations) from a MCMCglmm object

Usage

```
  ## S3 method for class 'postMCMCglmmRE'
print(x, ...)
```

Arguments

x	An postMCMCglmmRE object
...	Not currently used.

Value

Nothing. Prints to console.

See Also

stdranef

Examples

```
## Not run:
  # make me!!!

## End(Not run)
```

---

ranef.MCMCglmm   *Extract random effects from an* MCMCglmm *object*

Description

Function designed to extract the random effects from an MCMCglmm model object. Can either extract all samples from the random effects posteriors or return the posterior means.

Usage

```
  ## S3 method for class 'MCMCglmm'
 ranef(object, use = c("all", "mean"),
    ...)
```

Arguments

object	An MCMCglmm model object to extract the effects from
use	A character string indicating whether to extract all posterior samples or the mean of the posteriors. Defaults to "all".
...	Arguments passed on to the worker function.

Value

A matrix of the fixed effects

See Also

fixef.MCMCglmm

Examples

```
## Not run:
  # a simple MCMCglmm model
  data(PlodiaPO)
  m <- MCMCglmm(PO ~ 1, random= ~ FSfamily, data=PlodiaPO, pr=TRUE, verbose=FALSE)

  # only extract average fixed effects
  head(ranef(m, use = "mean"))

  # histogram of posterior samples of fixed effects
  hist(ranef(m)[1, ])
  # matches the mean
```

```
    rowMeans(ranef(m)[1:6, ])

  ## End(Not run)
```

---

ranefLevels	*Extract the levels of factors used for random effects in* MCMCglmm *objects*

Description

Extract the levels of factors used for random effects in MCMCglmm objects

Usage

```
    ranefLevels(object, data, ...)
```

Arguments

object	An MCMCglmm model object
data	The dataset used for the model
...	Not currently used

See Also

[paramNamesMCMCglmm](), [ranef.MCMCglmm]()

Examples

```
## Not run:
  # a simple MCMCglmm model
  data(PlodiaPO)
  m <- MCMCglmm(PO ~ 1, random = ~ FSfamily, data = PlodiaPO, verbose=FALSE)

  # extract the random effects levels
  ranefLevels(m, PlodiaPO)

## End(Not run)
```

---

recycler	*Calculate change in predicted probabilities*

Description

recycler wraps many of the functions in **postMCMCglmm** to calculate the change in predicted probabilities for a twiddle change in the predictor, or for discrete predictors, it can use values so it is the change from 0 to 1 (for example). The result is a MCMCglmmPredictedProbs (of course a difference but still) object, so it can be summarized using the MCMCglmmPredictedProbs summary method. This gives average marginal recycled predicted probabilities, as well as highest posterior density intervals.

Usage

```
recycler(object, index = 2L, twiddle, values, ...)
```

Arguments

`object`	A `MCMCglmm` model object to use for recycled predictions.
`index`	An integer indicating the column of the fixed effects design matrix, X, to vary. Defaults to 2L.
`twiddle`	A twiddle value for continuous variables. Needs to be small enough for the scale of the predictor that a twiddle change is a reasonable approximation of taking the first derivative at a point. That is, a very small change. If missing, reverts to .01.
`values`	Specific values to use for the varying predictor. These are primarily meant for discrete predictors rather than continuous ones.
`...`	Passed on to `predict2`

Value

A list of class `MCMCglmmPredictedProbs` that are the differences in predicted probabilities for a one unit change (calculated from the twiddle value or between the discrete values supplied in `values`).

See Also

[`summary.MCMCglmmPredictedProbs`](), [`predict2.MCMCglmm`]()

Examples

```
## Not run:
  ## Make me!

## End(Not run)
```

---

stdranef	*Extract standard deviation of "random" effects from an* `MCMCglmm` *object*

Description

Function designed to extract the standard deviation of the random effects from an `MCMCglmm` model object. Note that this is not the same as the posterior distribution of (co)variance matrices. It is based on the posterior distribution of the random effects. This also means it requires `pr=TRUE` to be set in the model for the information to be saved. Can optionally return standard deviation of random effects after back transforming to the response metric. Currently probabilities, but only for ordinal family models (`family="ordinal"`).

Usage

```
stdranef(object, which, type = c("lp", "response"), ...)
```

Arguments

`object`	An `MCMCglmm` model object to extract the effects from
`which`	A list of random effects to extract or their numeric positions If there are two numbers in a list, effects are simulataneous.
`type`	A chacter string indicating whether to calculate the standard deviation on the linear predictor metric, 'lp' or response, 'response'.
`...`	Not currently used.

Value

A list of class postMCMCglmmRE with means (`M`) and individual estimates (`Data`)

See Also

`print.postMCMCglmmRE`, `predict2.MCMCglmm`, `ranef.MCMCglmm`

Examples

```
## Not run:
  # a simple MCMCglmm model
  data(PlodiaPO)
  PlodiaPO <- within(PlodiaPO, {
    PO2 <- cut(PO, quantile(PO, c(0, .33, .66, 1)))
    plate <- factor(plate)
  })

  m <- MCMCglmm(PO2 ~ 1, random = ~ FSfamily + plate,
    family = "ordinal", data = PlodiaPO,
    prior = list(
      R = list(V = 1, fix = 1),
      G = list(
        G1 = list(V = 1, nu = .002),
        G2 = list(V = 1, nu = .002)
      )
    ), verbose=FALSE, thin=1, pr=TRUE)

  # summary of the model
  summary(m)

  # examples of extracting standard deviations of
  # different random effects on the linear predictor metric
  # or after transformation to probabilities (only for ordinal)
  stdranef(m, which = list(1), type = "lp")
  stdranef(m, which = list(2), type = "lp")
  stdranef(m, which = list(1, 2, c(1, 2)), type = "lp")
  stdranef(m, type = "lp")

  ## error because no 3rd random effect
  #stdranef(m, which = list(1, 2, 3), type = "lp")

  stdranef(m, which = list("FSfamily", "plate"), type = "lp")

  # mean standard deviations on the probability metric
  # also the full distributions, if desired in the Data slot.
  res <- stdranef(m, type = "response")
```

```
    res$M # means
    hist(res$Data$FSfamily[, 1]) # histogram

  ## End(Not run)
```

---

summary.MCMCglmmPredictedLP

Summary method for MCMCglmm predicted values

Description

If the predicted values only used the posterior means, highest posterior density (HPD) intervals cannot be generated, so only the means are returned. Otherwise, it calculates the mean predicted value, as well as the HPD interval.

Usage

```
   ## S3 method for class 'MCMCglmmPredictedLP'
 summary(object,
    level = 0.95, ...)
```

Arguments

object	A MCMCglmmPredictedLP object to summarize
level	A numeric value, the value to use when calculating HPD intervals. Defaults to .95.
...	Not currently used.

Value

If HPD intervals are calculated, returns a matrix with the means, lower limit, and upper limit. If no HPD intervals are calculated, returns a column vector of means.

See Also

[predict2.MCMCglmm](), [recycler]()

Examples

```
## Not run:
  ## Make me!

## End(Not run)
```

summary.MCMCglmmPredictedProbs

Summary method for MCMCglmm predicted probabilities

Description

Optionally first marginalizes across all observations by taking the row means. If the predicted values only used the posterior means, highest posterior density (HPD) intervals cannot be generated, so only the means are returned. Otherwise, it calculates the mean predicted probability, as well as the HPD interval. This can either be per observation or marginalized.

Usage

```
 ## S3 method for class 'MCMCglmmPredictedProbs'
summary(object,
   marginalize = FALSE, level = 0.95, ...)
```

Arguments

object	A `MCMCglmmPredictedProbs` object to summarize
marginalize	Logical whether or not to marginalize by taking the row means. Defaults to `FALSE`.
level	A numeric value, the value to use when calculating HPD intervals. Defaults to .95.
...	Not currently used.

Value

If HPD intervals are calculated, returns a list of matrices with the means, lower limit, and upper limit. If no HPD intervals are calculated, returns a list of means.

See Also

[predict2.MCMCglmm](#), [recycler](#)

Examples

```
## Not run:
  ## Make me!

## End(Not run)
```

Index