Package 'Ratings'

April 17, 2009

Version 0.1-1

Date 2008-08-05

Title Model-based Ratings Figures

Author Kevin M. Quinn <kevin_quinn@harvard.edu>, and Daniel E. Ho <dho@law.stanford.edu>

Maintainer Kevin M. Quinn <kevin_quinn@harvard.edu>

Depends R (>= 2.2.0), coda (>= 0.10-3)

Description This package contains functions to implement the methods described in Ho and Quinn. n.d. "Improving the Presentation and Interpretation of Online Ratings Data with Model-based Figures". The American Statistician.

License GPL-2

URL none

Repository CRAN

Date/Publication 2008-08-31 11:01:49

R topics documented:

`barplotModelBased` *Model-based Barplots of Ratings Data*

Description

Creates the model-based barplot described in Ho and Quinn (n.d.).

Usage

```
barplotModelBased(tau.mat, barcol = "darkgray", top.limit = 0.5,
                  scale.factor = 1, ...)
```

Arguments

`tau.mat`	A $P \times C$ matrix where P is the number of products being rated and C is the number of rating categories. The pth row, cth column of `tau.mat` should give the probability that product p is equal to category c.
`barcol`	The color for the bars in the barplot.
`top.limit`	The maximum value of the y axis. Should be between 0 and 1.
`scale.factor`	Scaling factor that adjusts the scaling of the y axis. Any positive value is allowed.
`...`	Other arguments to plotting functions.

References

Ho, Daniel E. and Kevin M. Quinn. forthcoming. "Improving the Presentation and Interpretation of Online Ratings Data with Model-based Figures." The American Statistician.

See Also

`ordrating`, `tauCalculate`, `starplotModelBased`

Examples

```
## Not run:
## Mondo Times example from Ho & Quinn (nd).
## may have to increase stack limit to run this example on some machines

data(Mondo)

ord.out <- ordrating(Mondo, beta.constraint=1, tune=.035,
                     ma=1, mb=-5, vinva=1, vinvb=0.05,
                     gamma.start=c(-300, 0, 1.5, 3.0, 4.5, 300),
                     thin=20, burnin=20000, mcmc=100000, verbose=1000)

## get rating probabilities
tau <- tauCalculate(ord.out, 500)
```

```
## just the labeled outlets
tau.sub <- tau[-grep("thetaOutlet", rownames(tau)),]

## clean up names
rownames(tau.sub) <-  gsub("theta", "", rownames(tau.sub))

barplotModelBased(tau.sub)

## more informative labels
colnames(tau.sub) <- c("Awful", "Poor", "Average", "Very Good", "Great")

barplotModelBased(tau.sub)



## subsetting the Mondo data to include only raters who rated 5 or more
## outlets (should avoid any stacksize problems)

Mondo.sub <- Mondo[apply(!is.na(Mondo), 1, sum) >= 5, ]
## also getting rid of outlets that are not rated now
Mondo.sub <- Mondo.sub[,apply(is.na(Mondo.sub), 2, mean) != 1]

ord.out <- ordrating(Mondo.sub, beta.constraint=1, tune=.035,
                     ma=1, mb=-5, vinva=1, vinvb=0.05,
                     gamma.start=c(-300, 0, 1.5, 3.0, 4.5, 300),
                     thin=20, burnin=20000, mcmc=100000, verbose=1000)

## get rating probabilities
tau <- tauCalculate(ord.out, 500)

## just the labeled outlets
tau.sub <- tau[-grep("thetaOutlet", rownames(tau)),]

## clean up names
rownames(tau.sub) <-  gsub("theta", "", rownames(tau.sub))

barplotModelBased(tau.sub)

## more informative labels
colnames(tau.sub) <- c("Awful", "Poor", "Average", "Very Good", "Great")

barplotModelBased(tau.sub)

## End(Not run)
```

`barplotObserved` *Observed Data Barplots of Ratings Data*

Description

Creates the observed data barplot discussed in Ho and Quinn (n.d.).

Usage

```
barplotObserved(Y, catnames, barcol = "darkgray", top.limit = 0.5,
                scale.factor = 1, ...)
```

Arguments

`Y`	Matrix of data to be analyzed. Entries must be integers from $1, \ldots, C$ or `NA` where C is the number of ordinal categories. Raters are on the rows and products are on the columns.
`catnames`	A character vector of names for the rating categories. Should have C elements. First element should give the name for a rating of 1, second element the names for a rating of 2, etc.
`barcol`	The color for the bars in the barplot.
`top.limit`	The maximum value of the y axis. Should be between 0 and 1.
`scale.factor`	Scaling factor that adjusts the scaling of the y axis. Any positive value is allowed.
`...`	Other arguments to plotting functions.

References

Ho, Daniel E. and Kevin M. Quinn. forthcoming. "Improving the Presentation and Interpretation of Online Ratings Data with Model-based Figures." The American Statistician.

See Also

`ordrating`, `barplotModelBased`, `starplotObserved`

Examples

```
## Not run:
## Mondo Times example from Ho & Quinn (nd).

data(Mondo)

## just the first 12 cols
Mondo.sub <- Mondo[,1:12]

barplotObserved(Mondo.sub, top.limit=1, scale.factor=.7,
                catnames = c("Awful", "Poor", "Average",
                             "Very Good", "Great"))

## End(Not run)
```

Mondo *Mondo Times News Media Content Quality Data*

Description

This data matrix contains ratings of the content quality of news media as collected by Mondo Times.

Usage

```
data(Mondo)
```

Format

Rows of this data matrix correspond to raters and columns to news outlets. The entry in the ith row and jth columns corresponds to the rating given by rater i to news outlet j. These ratings are on a five point scale: 1 = "Awful", 2 = "Poor", 3 = "Average", 4 = "Very Good", and 5 = "Great". NA values indicate that the rater in question did not rate the news outlet in question.

The first 12 columns of the data matrix have the full names of the news outlets so that the analysis in Ho and Quinn (forthcoming) can be replicated. The remaining news outlets are not named due to a confidentiality agreement with Mondo Times.

Source

Mondo Times http://www.mondotimes.com/

References

Ho, Daniel E. and Kevin M. Quinn. forthcoming. "Improving the Presentation and Interpretation of Online Ratings Data with Model-based Figures." The American Statistician.

Examples

```
data(Mondo)
```

ordrating *MCMC for Unidimensional Ordinal IRT Model*

Description

This function generates a sample from the posterior distribution of a unidimensional ordinal item response theory (IRT) model, with Gaussian priors on the ability and item parameters. The user supplies data and priors, and a sample from the posterior distribution is returned as an mcmc object, which can be subsequently analyzed with functions provided in the coda package.

Usage

```
ordrating(Y, beta.constraint = NULL, theta.neg.index = NULL,
             theta.pos.index = NULL, vinva = 0.2, vinvb = 0.2, ma = 0,
             mb = 1, theta.start = NULL, gamma.start = NULL,
             burnin = 1000, mcmc = 10000, thin = 1, tune = 1,
             verbose = 0, seed = NA)
```

Arguments

`Y`	Matrix of data to be analyzed. Entries must be integers from $1, \ldots, C$ or `NA` where C is the number of ordinal categories. Items are on the rows and subjects are on the columns.
`beta.constraint`	Possible inequality constraint on all β parameters. `beta.constraint = NULL` (default) implies no constraint, `beta.constraint > 0` implies all β values are constrained to be positive, and `beta.constraint < 0` implies all β values are constrained to be negative.
`theta.neg.index`	An index of θ that is constrained to be negative. To constrain the θ for the subject whose data is in the jth column of `Y` to be negative one would set `theta.neg.index = j`.
`theta.pos.index`	An index of θ that is constrained to be positive. To constrain the θ for the subject whose data is in the jth column of `Y` to be positive one would set `theta.pos.index = j`.
`vinva`	The prior inverse variance for each α_r.
`vinvb`	The prior inverse variance for each β_r.
`ma`	The prior mean for each α_r.
`mb`	The prior mean for each β_r.
`theta.start`	Starting values for θ. If `NULL` starting values are calculated based on the sample means of `Y`.
`gamma.start`	Starting values for γ. If `NULL` starting values are chosen to be equally spaced between 1 and the number of ordinal categories C.
`burnin`	The number of burn-in iterations for the sampler.
`mcmc`	The number of MCMC iterations for the sampler.
`thin`	The thinning interval used in the simulation. The number of Gibbs iterations must be divisible by this value.
`tune`	The scalar tuning parameter for the Metropolis-Hastings sampling. Must be strictly positive.
`verbose`	A switch which determines whether or not the progress of the sampler is printed to the screen. If `verbose` is greater than 0 then every `verbose`th iteration will be printed to the screen.

`seed` The seed for the random number generator. If NA, the Mersenne Twister generator is used with default seed 12345; if an integer is passed it is used to seed the Mersenne twister. The user can also pass a list of length two to use the L'Ecuyer random number generator, which is suitable for parallel computation. The first element of the list is the L'Ecuyer seed, which is a vector of length six or NA (if NA a default seed of `rep(12345,6)` is used). The second element of list is a positive substream number. See the MCMCpack specification for more details.

Details

Let $r = 1, \ldots, R$ index items, $p = 1, \ldots, P$ index subjects, and $c = 1, \ldots, C$ index ordinal rating categories.

Y is an $R \times P$ matrix with elements in $\{1, \ldots, C, NA\}$. `NA` denotes missing data that are assumed to be missing at random (MAR).

The distribution of Y is governed by a $R \times P$ matrix of latent variables Y^* and a series of cutpoints γ.

More specifically,

$$y^*_{rp} = \alpha_r + \beta_r \theta_p + \epsilon_{rp}$$

where $\epsilon_{rp} \sim N(0, 1)$. It is assumed that $y_{rp} = c$ if and only if

$$y^*_{rp} \in (\gamma_{c-1}, \gamma_c]$$

.

The priors used for this model are that each α_r is iid Gaussian, each β_r is iid Gaussian, each θ_p is standard normal, and the elements of γ are improper uniform with all parameters assumed to be a priori independent.

`ordrating` simulates from the posterior distribution using a Metropolis-Hastings within Gibbs sampling algorithm. The algorithm employed is based on work by Cowles (1996). Note that the first element γ_1 is normalized to zero, and thus not returned in the mcmc object.

`ordrating` fits a model that is a special case of the model fit by `MCMCordfactanal` in the `MCMCpack` package. The primary differences are the types of identifying constraints employed, the dimensionality of the θ and β parameters, and the computational speed. Because `ordrating` fits a narrower class of models it can be optimized for speed much more effectively.

Value

An mcmc object that contains the posterior sample. This object can be summarized by functions provided by the `coda` package.

References

Ho, Daniel E. and Kevin M. Quinn. forthcoming. "Improving the Presentation and Interpretation of Online Ratings Data with Model-based Figures." The American Statistician.

M. K. Cowles. 1996. "Accelerating Monte Carlo Markov Chain Convergence for Cumulative-link Generalized Linear Models." *Statistics and Computing.* 6: 101-110.

Valen E. Johnson and James H. Albert. 1999. "Ordinal Data Modeling." Springer: New York.

Kevin M. Quinn. 2004. "Bayesian Factor Analysis for Mixed Ordinal and Continuous Responses." *Political Analysis*. 12: 338-353.

Shawn Treier and Simon Jackman. 2003. "Democracy as a Latent Variable." Paper presented at the Midwest Political Science Association Annual Meeting.

Martyn Plummer, Nicky Best, Kate Cowles, and Karen Vines. 2002. *Output Analysis and Diagnostics for MCMC (CODA)*. http://www-fis.iarc.fr/coda/.

See Also

plot.mcmc,summary.mcmc, MCMCordfactanal

Examples

```
## Not run:
## Mondo Times example from Ho & Quinn (nd).
## may have to increase stack limit to run this example on some machines

data(Mondo)

ord.out <- ordrating(Mondo, beta.constraint=1, tune=.035,
                     ma=1, mb=-5, vinva=1, vinvb=0.05,
                     gamma.start=c(-300, 0, 1.5, 3.0, 4.5, 300),
                     thin=20, burnin=20000, mcmc=100000, verbose=1000)

plot(ord.out)
summary(ord.out)



## subsetting the Mondo data to include only raters who rated 5 or more
## outlets (should avoid any stacksize problems)

Mondo.sub <- Mondo[apply(!is.na(Mondo), 1, sum) >= 5, ]
## also getting rid of outlets that are not rated now
Mondo.sub <- Mondo.sub[,apply(is.na(Mondo.sub), 2, mean) != 1]

ord.out <- ordrating(Mondo.sub, beta.constraint=1, tune=.035,
                     ma=1, mb=-5, vinva=1, vinvb=0.05,
                     gamma.start=c(-300, 0, 1.5, 3.0, 4.5, 300),
                     thin=20, burnin=20000, mcmc=100000, verbose=1000)

plot(ord.out)
summary(ord.out)

## End(Not run)
```

`starplotModelBased` *Model-based Starplots of Ratings Data*

Description

Creates the model-based starplot described in Ho and Quinn (n.d.).

Usage

```
starplotModelBased(tau.mat, colvec = NULL, starsize = 0.2,
                   interpolation.level = 200, ...)
```

Arguments

`tau.mat`	A $P \times C$ matrix where P is the number of products being rated and C is the number of rating categories. The pth row, cth column of `tau.mat` should give the probability that product p is equal to category c.
`colvec`	Vector of rgb colors used for the plot.
`starsize`	Positive scalar that regulates the size of the stars in the plot.
`interpolation.level`	Regulates the smoothness of the color scale. `interpolation.level = 1` simply uses the colors in `colvec`. Values of `interpolation.level > 1` provide increasing amounts of interpolation between the values in `colvec`. Larger values of `interpolation.leve` produce smoother transitions between colors.
`...`	Other arguments to plotting functions.

References

Ho, Daniel E. and Kevin M. Quinn. forthcoming. "Improving the Presentation and Interpretation of Online Ratings Data with Model-based Figures." The American Statistician.

See Also

`ordrating`, `tauCalculate`, `barplotModelBased`

Examples

```
## Not run:
## Mondo Times example from Ho & Quinn (nd).
## may have to increase stack limit to run this example on some machines

data(Mondo)

ord.out <- ordrating(Mondo, beta.constraint=1, tune=.035,
                     ma=1, mb=-5, vinva=1, vinvb=0.05,
                     gamma.start=c(-300, 0, 1.5, 3.0, 4.5, 300),
```

```
                    thin=20, burnin=20000, mcmc=100000, verbose=1000)

## get rating probabilities
tau <- tauCalculate(ord.out, 500)

## just the labeled outlets
tau.sub <- tau[-grep("thetaOutlet", rownames(tau)),]

## clean up names
rownames(tau.sub) <-  gsub("theta", "", rownames(tau.sub))

starplotModelBased(tau.sub)

## more informative labels
colnames(tau.sub) <- c("Awful", "Poor", "Average", "Very Good", "Great")

starplotModelBased(tau.sub)



## subsetting the Mondo data to include only raters who rated 5 or more
## outlets (should avoid any stacksize problems)

Mondo.sub <- Mondo[apply(!is.na(Mondo), 1, sum) >= 5, ]
## also getting rid of outlets that are not rated now
Mondo.sub <- Mondo.sub[,apply(is.na(Mondo.sub), 2, mean) != 1]

ord.out <- ordrating(Mondo.sub, beta.constraint=1, tune=.035,
                     ma=1, mb=-5, vinva=1, vinvb=0.05,
                     gamma.start=c(-300, 0, 1.5, 3.0, 4.5, 300),
                     thin=20, burnin=20000, mcmc=100000, verbose=1000)

## get rating probabilities
tau <- tauCalculate(ord.out, 500)

## just the labeled outlets
tau.sub <- tau[-grep("thetaOutlet", rownames(tau)),]

## clean up names
rownames(tau.sub) <-  gsub("theta", "", rownames(tau.sub))

starplotModelBased(tau.sub)

## more informative labels
colnames(tau.sub) <- c("Awful", "Poor", "Average", "Very Good", "Great")

starplotModelBased(tau.sub)

## a different color scheme
mycol <- rgb(red=c(255, 243, 231, 219, 207, 159, 82.5, 30, 0, 0, 255),
             green=c(255.0, 250.1, 245.2, 240.3, 235.4, 215.8, 175, 144,
               0, 0, 69),
             blue=c(255, 255, 255, 255, 255, 255, 255, 255, 238, 183, 0),
```

```
                maxColorValue=255
                )

starplotModelBased(tau.sub, colvec=mycol)

## End(Not run)
```

`starplotObserved` *Observed Data Starplots of Ratings Data*

Description

Creates the observed data starplot discussed in Ho and Quinn (n.d.).

Usage

```
starplotObserved(Y, catnames, starcol = "darkgray", starsize = 0.2, ...)
```

Arguments

`Y`	Matrix of data to be analyzed. Entries must be integers from $1, \ldots, C$ or `NA` where C is the number of ordinal categories. Raters are on the rows and products are on the columns.
`catnames`	A character vector of names for the rating categories. Should have C elements. First element should give the name for a rating of 1, second element the names for a rating of 2, etc.
`starcol`	Color of stars.
`starsize`	Positive scalar that regulates the size of the stars in the plot.
`...`	Other arguments to plotting functions.

References

Ho, Daniel E. and Kevin M. Quinn. forthcoming. "Improving the Presentation and Interpretation of Online Ratings Data with Model-based Figures." The American Statistician.

See Also

`ordrating`, `starplotModelBased`, `barplotObserved`

Examples

```
## Not run:
## Mondo Times example from Ho & Quinn (nd).

data(Mondo)

## just the first 12 cols
Mondo.sub <- Mondo[,1:12]
```

```
starplotObserved(Mondo.sub, starcol="green",
                  catnames = c("Awful", "Poor", "Average",
                               "Very Good", "Great"))

## End(Not run)
```

`tauCalculate` *Posterior Predictive Probabilities from ordrating*

Description

Calculates the quantity labeled τ_{pc} in Ho and Quinn (n.d.). τ_{pc} can be thought of as the probability that a randomly chosen rater (from the set of observed raters) will give product p a rating of c given the observed data.

Usage

```
tauCalculate(out, ndraws = 500)
```

Arguments

`out`	An output object from the `ordrating` function.
`ndraws`	The number of Monte Carlo draws used to calculate the posterior predictive probabilities. Must be less than or equal to the number of rows in `out`.

Details

Section 3.2 of Ho and Quinn (n.d.) provides full details for how τ_{pc} is calculated. The basic ideas are the following.

`tauCalculate` takes the MCMC output from `ordrating` and calculates the sample average (over all row units in Y) of the posterior predictive probability of a particular column unit in Y being rated as c. This is done for all column units and ratings categories.

Value

A $P \times C$ matrix where P is the number of columns in Y and C is the number of ordinal rating categories. Each row of this matrix gives the posterior predictive probability that a randomly chosen rater, from the set of observed raters, will give product p a rating of c.

References

Ho, Daniel E. and Kevin M. Quinn. forthcoming. "Improving the Presentation and Interpretation of Online Ratings Data with Model-based Figures." The American Statistician.

See Also

`ordrating`, `barplotModelBased`, `starplotModelBased`

Examples

```
## Not run:
## Mondo Times example from Ho & Quinn (nd).
## may have to increase stack limit to run this example on some machines

data(Mondo)

ord.out <- ordrating(Mondo, beta.constraint=1, tune=.035,
                     ma=1, mb=-5, vinva=1, vinvb=0.05,
                     gamma.start=c(-300, 0, 1.5, 3.0, 4.5, 300),
                     thin=20, burnin=20000, mcmc=100000, verbose=1000)

tau <- tauCalculate(ord.out, 500)



## subsetting the Mondo data to include only raters who rated 5 or more
## outlets (should avoid any stacksize problems)

Mondo.sub <- Mondo[apply(!is.na(Mondo), 1, sum) >= 5, ]
## also getting rid of outlets that are not rated now
Mondo.sub <- Mondo.sub[,apply(is.na(Mondo.sub), 2, mean) != 1]

ord.out <- ordrating(Mondo.sub, beta.constraint=1, tune=.035,
                     ma=1, mb=-5, vinva=1, vinvb=0.05,
                     gamma.start=c(-300, 0, 1.5, 3.0, 4.5, 300),
                     thin=20, burnin=20000, mcmc=100000, verbose=1000)

tau <- tauCalculate(ord.out, 500)

## End(Not run)
```

Index