

Package: speedyBBT (via r-universe)

November 7, 2024

Title Efficient Bayesian Inference for the Bradley--Terry Model

Version 1.0.0.9000

Description A suite of functions that allow a full, fast, and efficient Bayesian treatment of the Bradley--Terry model. Prior assumptions about the model parameters can be encoded through a multivariate normal prior distribution. Inference is performed using a latent variable representation of the model.

License GPL (>= 3)

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Imports base, BayesLogit, Matrix, stats, utils

Suggests expm, testthat (>= 3.0.0)

Depends R (>= 2.10)

Config/testthat/edition 3

Repository <https://rowlandseymour.r-universe.dev>

RemoteUrl <https://github.com/rowlandseymour/speedybbt>

RemoteRef HEAD

RemoteSha 7f58ede8ca23961e001b97082b9ba5de29310126

Contents

BBTm	2
BBTm.ties	4
comparisons_to_matrix	6
darEsSalaam	6
forcedMarriage	7
oxon.comparisons	8
speedyBBTm	8
sy.comparisons	10
wimbledon	10

BBTm

*Generalised Bradley-Terry model***Description**

This function fits the Bradley-Terry model with comparison and player specific effects. Each comparison can be assigned a real value to allow for a specific effect for the comparison, such as bias, ordering or home/away effect. The value of this effect is denoted kappa. The player specific effects are described through a formula and data.frame containing the value. The function places a normal prior distribution on both kappa and the player specific parameters beta.

Usage

```
BBTm(
  outcome,
  player1,
  player2,
  lambda.initial = NULL,
  player.prior.var = NULL,
  beta.initial = NULL,
  n.iter = 1000,
  formula = NULL,
  data = NULL,
  advantage = NULL,
  kappa.initial = NULL,
  kappa.var = NULL,
  hyperparameter = TRUE,
  chi = 0.01,
  psi = 0.01
)
```

Arguments

outcome	vector of outcomes. 1 if player2 is the winner, 0 if player1 is the winner
player1	vector of first players.
player2	vector of second players.
lambda.initial	(optional) vector containing the values of the player parameters for the first MCMC iteration
player.prior.var	(optional) matrix specifying the prior covariance of the player correlation parameters
beta.initial	(optional) vector containing the values of the player specific parameters for the first MCMC iteration
n.iter	number of MCMC samples to be drawn

formula	formula with no left-hand-side specifying the player specific effects
data	data.frame with a row corresponding to each player and column corresponding to each covariate.
advantage	(optional) a vector with the value of the comparisons specific effect for each comparison
kappa.initial	(optional) an initial value for the comparison specific value kappa
kappa.var	(optional) the prior variance of the he comparison specific value kappa
hyperparameter	boolean indicating if inference should be performed for the prior variance hyperparameter. If TRUE the prior variance (main diagonal of the covariance matrix) must be set to 1.
chi	rate parameter for the inverse-gamma prior distribution on the hyperparameter
psi	shape parameter for the inverse-gamma prior distribution on the hyperparameter

Details

If `player.prior.var` is omitted, independent and identical $N(0, 5^2)$ prior distributions are placed on each object quality parameter.

If `beta.initialis` omitted, it is set to a vector of zeroes.

If `kappa.var` is omitted, it is set to $N(0, 5^2)$, if `kappa.initial` is omitted it is set to 0.5.

Value

A data frame containing samples from the posterior distribution

Examples

```
#####
## Wimbledon 2019 ##
#####

#Fit model where the quality of each player depends on their rank
#and the number of points they had immediately before the tournament.
#Allow an effect for a match being in the first or second week.
#wimbledonModel <- BBTm(outcome = wimbledon$matches$outcome,
#                          player2 = wimbledon$matches$loser,
#                          player1 = wimbledon$matches$winner,
#                          advantage = wimbledon$matches$secondWeek,
#                          formula = ~ rank + points,
#                          data = wimbledon$players,
#                          n.iter = 4000)

#Plot posterior distributions
#hist(wimbledonModel$kappa[-c(1:100)], main = "", xlab = expression(kappa), freq = FALSE)
#hist(wimbledonModel$beta[-c(1:100), 1], main = "", xlab = expression(beta[1]), freq = FALSE)
#hist(wimbledonModel$beta[-c(1:100), 2], main = "", xlab = expression(beta[2]), freq = FALSE)
```

BBTm.ties

Bayesian inference for the Bradley–Terry model with ties

Description

This function uses MCMC to sample from the posterior distribution of the Bradley–Terry model with ties. A multivariate normal prior distribution on the player quality parameters can be specified. An exponential prior distribution is placed on the tie parameter θ , and a Metropolis–Hasting random walk algorithm is used to update this parameter.

Usage

```
BBTm.ties(
  n.objects,
  outcome,
  player1,
  player2,
  player.prior.var = NULL,
  theta.initial = NULL,
  lambda.initial = NULL,
  n.iter = 1000,
  hyperparameter = TRUE,
  chi = 0.01,
  psi = 0.01,
  rw.sd = 0.1,
  theta.rate = 0.01
)
```

Arguments

n.objects	number of objects in the study
outcome	vector of outcomes. 0 if player 1 is the winner, 1 if player 2 is the winner, and 2 if it is a tie.
player1	vector of first players.
player2	vector of second players.
player.prior.var	(optional) matrix specifying the prior covariance of the player correlation parameters
theta.initial	(optional) value of the tied parameter there for the first MCMC iteration
lambda.initial	(optional) vector containing the values of the player parameters for the first MCMC iteration
n.iter	number of MCMC samples to be drawn
hyperparameter	boolean indicating if inference should be performed for the prior variance hyperparameter. If TRUE the prior variance (main diagonal of the covariance matrix) must be set to 1.

chi	rate parameter for the inverse-gamma prior distribution on the hyperparameter
psi	shape parameter for the inverse-gamma prior distribution on the hyperparameter
rw.sd	number describing the standard deviation of normal distribution proposal distribution for theta
theta.rate	(optional) The rate parameter of the exponential prior distribution placed on theta

Details

If `player.prior.var` is omitted, independent and identical $N(0, 5^2)$ prior distributions are placed on each object quality parameter.

If `lambda.initial` is omitted, it is set to a vector of zeroes.

Value

A data frame containing samples from the posterior distribution

Examples

```
#####
## Deprivation in Dar es Salaam, Tanzania ##
## Seymour et al (2022) ##
#####

#Construct covariance matrix based on spatial informartion
sigma <- expm::expm(darEsSalaam$adjacencyMatrix)
sigma <- diag(diag(sigma)^-0.5)%*% sigma %*%diag(diag(sigma)^-0.5)

##Not Run

#Fit BT model with ties
#darTiedModel <- BBTm.ties(n.objects = 452,
#                           outcome = darEsSalaam$comparisons$outcome,
#                           player1 = darEsSalaam$comparisons$subward1,
#                           player2 = darEsSalaam$comparisons$subward2,
#                           player.prior.var = sigma,
#                           hyperparameter = TRUE, rw.sd = 0.005)

#Get posterior means
#darTiedModel$lambda <- darTiedModel $lambda - colMeans(darTiedModel$lambda)
#lambda.mean <- rowMeans(darTiedModel$lambda)

#Generate trace plots
#plot(lambda.mean)
#plot(darTiedModel$theta[-c(1:100)], type = 'l')
```

comparisons_to_matrix *Construct Win Matrix from Comparisons*

Description

This function constructs a win matrix from a data frame of comparisons. It is needed for the MCMC functions.

Usage

```
comparisons_to_matrix(n.objects, comparisons)
```

Arguments

n.objects	The number of areas in the study.
comparisons	An N x 2 data frame, where N is the number of comparisons. Each row should correspond to a judgment. The first column is the winning object, the second column is the more losing object. The areas should be labeled from 1 to n.objects.

Value

A matrix where the i, j th element is the number of times object i beat object j .

Examples

```
#Generate some sample comparisons
comparisons <- data.frame("winner" = c(1, 3, 2, 2), "loser" = c(3, 1, 1, 3))

#Create matrix from comparisons
win.matrix <- comparisons_to_matrix(3, comparisons)
```

darEsSalaam

Comparative Judgment on Deprivation in Dar es Salaam, Tanzania

Description

A comparative judgment data set on deprivation in subwards in Dar es Salaam, Tanzania. Citizens were shown pairs of subwards at random and asked which was more deprived. If they said they were equal, one of the pair was chosen at random to be more deprived. The data was collected in August 2018. The sex of each judge is also included.

Usage

```
darEsSalaam
```

Format

A list with three elements. The first is a dataframe containing the comparison. Each row corresponds to a judgement made by a single judge. Columns 2 and 3 contain the pair of subwards being compared. The first column shows the outcome of the comparison: 1 if player 2 won, 2 if it was a tie and 0 if player 1 won (although there are no instances of this happening). This differs from the data in the BSBT package as it explicitly includes ties rather than randomly allocating a winner.

The second is a dataframe containing the names and shapefiles of the subwards

The third is an adjacency matrix of the subwards formed from the shapefiles. This considers subwards as nodes and places edges between adjacent subwards. Two additional edges have been manually included to allow for crossings of the Kurasini creek.

Source

This data set was collected by Madeleine Ellis, James Goulding, Bertrand Perrat, Gavin Smith and Gregor Engelmann. We gratefully acknowledge the Rights Lab at the University of Nottingham for supporting funding for the comprehensive ground truth survey. We also acknowledge HumanitarianStreet Mapping Team (HOT) for providing a team of experts in data collection to facilitate the surveys. This work was also supported by the EPSRC Horizon Centre for Doctoral Training - My Life in Data (EP/L015463/1) and EPSRC grant Neodemographics (EP/L021080/1).

forcedMarriage

Forced Marriage in Nottinghamshire

Description

A comparative judgment data set for risk of forced marriage at ward level in Nottinghamshire. There are 12 judges and 76 wards.

Usage

forcedMarriage

Format

A list with three elements. The first is a dataframe containing 1846 rows and 4 columns. Each row corresponds to a judgement made by a single judge. Columns 3 and 4 show which of the pair of wards was judged to have relatively higher and low forced marriage risk level, column 1 shows which judge the comparison belongs to, and column 2 shows what time they made the decision.

The second is a dataframe describing each ward and its geometry.

The final element is an adjacency matrix, where the wards are nodes and edges are placed between adjacent wards.

@keywords datasets

@source The data was collected using support from the Engineering and Physical Sciences Research Council (grant reference EP/R513283/1), the Economic and Social Sciences Research Council (ES/V015370/1) and the Research England Policy Support Fund. The data was collected following ethical approval from the University of Nottingham School of Politics and International Relations ethics committee.

 oxon.comparisons

Honour Based Abuse in Oxfordshire

Description

A comparative judgment data set for risk of honour based abuse in Oxford and Banbury

Usage

oxon.comparisons

Format

A data frame with 1,167 comparisons. Each comparison has an ID, the ID of the user who made the comparisons, the IDs of the two areas involved in the comparisons, the ID of the selected area, and the state of the outcome. If the comparison was tied, the ID of the selected area is NA

@keywords datasets

@source The data was collected following ethical approval the University of Birmingham's Science, Engineering and Maths Ethics Committee.

 speedyBBTm

Standard Bayesian Bradley–Terry model

Description

This function uses MCMC to sample from the posterior distribution of the standard Bradley–Terry model. Standard model means that there are no tied comparisons and no player or comparison specific variables. This provides a fast implementation of the standard model. A multivariate normal prior distribution on the player quality parameters can be specified.

Usage

```
speedyBBTm(
  outcome = NULL,
  player1 = NULL,
  player2 = NULL,
  win.matrix = NULL,
  player.prior.var = NULL,
  lambda.initial = NULL,
  n.iter = 1000,
  hyperparameter = TRUE,
  chi = 0.01,
  psi = 0.01
)
```


Arguments

outcome	vector of outcomes. 1 if player 2 is the winner, 0 if player 1 is the winner
player1	vector of first players
player2	vector of second players
win.matrix	a win-loss matrix where the i,j th element is the number of times object i beat object j
player.prior.var	(optional) matrix specifying the prior covariance of the player correlation parameters
lambda.initial	(optional) vector containing the values of the player correlation parameters for the first MCMC iteration
n.iter	number of MCMC samples to be drawn
hyperparameter	boolean indicating if inference should be performed for the prior variance hyperparameter. If TRUE the prior variance (main diagonal of the covariance matrix) must be set to 1.
chi	rate parameter for the inverse-gamma prior distribution on the hyperparameter
psi	shape parameter for the inverse-gamma prior distribution on the hyperparameter

Details

If `player.prior.var` is omitted, independent and identical $N(0, 1^2)$ prior distributions are placed on each object quality parameter.

If `lambda.initial` is omitted, it is set to a vector of zeroes.

Value

A data frame containing samples from the posterior distribution

Examples

```
#####
## Forced Marriage in Nottinghamshire ##
#####

#Construct covariance matrix based on spatial information
sigma <- expm::expm(forcedMarriage$adjacencyMatrix)
sigma <- diag(diag(sigma)^-0.5)%*% sigma %*%diag(diag(sigma)^-0.5)

##Not Run
#Fit model
#forcedMarriageModel <- speedyBBTm(outcome = rep(1, length(forcedMarriage$comparisons$win)),
#                                   player1 = forcedMarriage$comparisons$win,
#                                   player2 = forcedMarriage$comparisons$lost,
#                                   player.prior.var = sigma)

#Plot results
```

```
#plot(sort(forcedMarriageQualitySamples))
```

sy.comparisons *FGM in South Yorkshire*

Description

A comparative judgment data set for risk of female genital mutilation at ward level in South Yorkshire.

Usage

```
sy.comparisons
```

Format

A data frame with 877 comparisons. Each comparison has an ID, the ID of the user who made the comparisons, the IDs of the two areas involved in the comparisons, the ID of the selected area, and the state of the outcome. If the comparison was tied, the ID of the selected area is NA

@keywords datasets

@source The data was collected following ethical approval the University of Birmingham's Science, Engineering and Maths Ethics Committee.

wimbledon *Wimbledon Men's Singles Championship 2019*

Description

The outcomes of all 127 men's singles matches in the 2019 Wimbledon championship.

Usage

```
wimbledon
```

Format

A list containing a dataframe with the outcomes of the matches and a dataframe describing the players. Each row of the matches dataframe corresponds to a match. The players dataframe has the name and id of the player as well as their rank in the ATP league table and the number of points received so far in the ATP 2019 tour prior to Wimbledon starting.

Source

<http://tennis-data.co.uk/alldata.php>

Index

* datasets

- darEsSalaam, [6](#)
- forcedMarriage, [7](#)
- oxon.comparisons, [8](#)
- sy.comparisons, [10](#)
- wimbledon, [10](#)

BBTm, [2](#)

BBTm.ties, [4](#)

comparisons_to_matrix, [6](#)

darEsSalaam, [6](#)

forcedMarriage, [7](#)

oxon.comparisons, [8](#)

speedyBBTm, [8](#)

sy.comparisons, [10](#)

wimbledon, [10](#)