

Package ‘sport’

January 8, 2019

Type Package

Title Sequential Pairwise Online Rating Techniques

Version 0.1.2

Depends R (>= 3.0)

Maintainer Dawid Kałedkowski <dawid.kaledkowski@gmail.com>

Description Calculates ratings for two-player or multi-player challenges. Methods included in package such as are able to estimate ratings (players strengths) and their evolution in time, also able to predict output of challenge. Algorithms are based on Bayesian Approximation Method, and they don't involve any matrix inversions nor likelihood estimation. Parameters are updated sequentially, and computation doesn't require any additional RAM to make estimation feasible. Additionally, base of the package is written in C++ what makes sport computation even faster. Methods used in the package refers to Mark E. Glickman (1999) <<http://www.glicko.net/research/glicko.pdf>>; Mark E. Glickman (2001) <[doi:10.1080/02664760120059219](https://doi.org/10.1080/02664760120059219)>; Ruby C. Weng, Chih-Jen Lin (2011) <<http://jmlr.csail.mit.edu/papers/volume12/weng11a/weng11a.pdf>>; W. Penny, Stephen J. Roberts (1999) <[doi:10.1109/IJCNN.1999.832603](https://doi.org/10.1109/IJCNN.1999.832603)>.

BugReports <https://github.com/gogonzo/sport/issues>

Imports Rcpp (>= 0.12.8), data.table (>= 1.11.0), ggplot2 (>= 3.0.0)

LinkingTo Rcpp

License GPL-2

Encoding UTF-8

URL <https://github.com/gogonzo/sport>

LazyData true

RoxygenNote 6.1.0

Suggests dplyr, knitr, magrittr, rmarkdown, testthat

VignetteBuilder knitr

NeedsCompilation yes

Author Dawid Kałedkowski [aut, cre]

Repository CRAN

Date/Publication 2019-01-07 23:00:10 UTC

R topics documented:

bbt_run	2
dbl_run	4
glicko2_run	5
glicko_run	7
gpheats	8
gpsquads	9
plot.rating	9
predict.rating	10
summary.rating	10

Index **12**

bbt_run	<i>BBT rating algorithm</i>
---------	-----------------------------

Description

BBT rating algorithm Wrapper around 'bbt' update algorithm. Wrapper allows user to simplify calculation providing only data and initial parameters assumptions

Usage

```
bbt_run(formula, data, r, rd, sigma, weight, beta = 25/6, kappa = 0.5,
        gamma, idlab, init_r = 25, init_rd = 25/3, pb = FALSE)
```

Arguments

formula	formula specifying model. BBT algorithm allows only player ranking parameter and should be specified by following manner: 'rank id ~ name'. Names in formula are unrestricted, but model structure remains the same: <ul style="list-style-type: none"> rank player position in event. id event identifier in which pairwise comparison is assessed. name of player.
data	data.frame which contains columns specified in formula, and optionally columns defined by 'sigma', 'weight' or 'date'.
r	named vector of initial rating estimates. In there is no assumption, initial ratings is set to be r=25 Names of vector should correspond with 'name' in formula.
rd	named vector of initial rating deviation estimates. In there is no assumption, initial is set to be r=25/3 Names of vector should correspond with 'name' in formula.

sigma	name of column in 'data' containing rating volatility. Rating volatility is a value which multiplies prior 'rd'. If 'sigma > 0' then prior 'rd' increases, making estimate of 'r' more uncertain.
weight	name of column in 'data' containing weights. Weights increasing or decreasing update change. Higher weight increasing impact of corresponding event.
beta	The additional variance of performance. As beta increases, the performance is more uncertain and update change is smaller. By default 'beta = 25/6'.
kappa	controls 'rd' shrinkage not to be greater than 'rd*(1-kappa)'. 'kappa=1' means that 'rd' will not be decreased.
gamma	can help to control how fast the variance 'rd' is reduced after updating. Lower 'gamma' slows decreasing of 'rd', which tends to reach zero to quickly. The default value is 'gamma = rd/c'.
idlab	name of column in 'data' containing date. Doesn't affect estimation process. If specified, charts displays estimates changes in time instead of by observation 'id'
init_r	initial values for 'r' if not provided. Default = 25
init_rd	initial values for 'r' if not provided. Default = 25/3
pb	logical, if TRUE progress bar will appear in console. Default = FALSE

Value

A "ratings" object is returned

- final_r named vector containing players ratings.
- final_rd named vector containing players ratings deviations.
- r data.frame with evolution of the ratings and ratings deviations estimated at each event.
- pairs pairwise combinations of players in analysed events with prior probability and result of a challenge.
- class of the object
- method type of algorithm used
- formula modelled formula

Examples

```
# Example from Glickman
data <- data.frame( name = c( "A", "B", "C", "D" ),
                   rank = c( 3, 4, 1, 2 ))
bbt <- bbt_run( rank ~ name, data )
```

dbl_run *DBL rating algorithm*

Description

DBL rating algorithm Wrapper around 'dbl' update algorithm. Wrapper allows user to simplify calculation providing only data and initial parameters assumptions

Usage

```
dbl_run(formula, data, r, rd, beta, weight, idlab, kappa = 0.5,
        init_r = 0, init_rd = 1, pb = FALSE)
```

Arguments

formula	formula specifying model. DBL allows multiple variables in formula, also two-way interaction are available specified by ':'. 'dbl' formula require first variable to be player name or identifier. LHS needs 'ranklid', to specify competitors order and event 'id'.
data	data.frame which contains columns specified in formula, and optionally columns defined by 'beta', 'weight' or 'date'.
r	named vector of initial estimates. If there is no assumption, initial ratings is set to be r=0.
rd	named vector of initial variance of 'r' estimates. In there is no assumption, initial is set to be rd=1.
beta	The additional variance of performance. As beta increases, the performance is more uncertain and update change is smaller. By default 'beta = 25/6'.
weight	name of column in 'data' containing weights. Weights increasing or decreasing update change. Higher weight increasing impact of corresponding event.
idlab	name of column in 'data' containing date. Doesn't affect estimation process. If specified, charts displays estimates changes in time in
kappa	parameter controlling 'rd' to avoid quick decreasing to zero. Is the proportion of 'rd' which is maximum change size.
init_r	initial values for 'r' if not provided. Default 'r=0'
init_rd	initial values for 'rd' if not provided. Default 'rd=1'
pb	logical, if TRUE progress bar will appear in console. Default = FALSE

Value

A "rating" object is returned

- final_r named vector containing players ratings.
- final_rd named vector containing players ratings deviations.
- r data.frame with evolution of the ratings and ratings deviations estimated at each event.

- pairs pairwise combinations of players in analysed events with prior probability and result of a challenge.
- class of the object
- method type of algorithm used
- formula modelled formula

Examples

```
data <- data.frame( name = c( "A", "B", "C", "D" ),
                   rank = c( 3, 4, 1, 2 ))
dbl <- dbl_run( rank ~ name, data)
```

glicko2_run	<i>Glicko2 rating algorithm</i>
-------------	---------------------------------

Description

Glicko2 rating algorithm Wrapper around ‘glicko2’ update algorithm. Wrapper allows user to simplify calculation providing only data and initial parameters assumptions

Usage

```
glicko2_run(formula, data, r, rd, sigma, tau = 0.5, weight,
            kappa = 0.5, idlab, init_r = 1500, init_rd = 350, pb = FALSE)
```

Arguments

formula	formula specifying model. Glicko2 algorithm allows only player ranking parameter and should be specified by following manner: ‘rank id ~ name’. Names in formula are unrestricted, but model structure remains the same: <ul style="list-style-type: none"> • rank player position in event. • id event identifier in which pairwise comparison is assessed. • name of player.
data	data.frame which contains columns specified in formula, and optionally columns defined by ‘weight’ or ‘date’.
r	named vector of initial rating estimates. In there is no assumption, initial ratings is set to be r=1500. Names of vector should correspond with team_name label.
rd	named vector of initial rating deviation estimates. In there is no assumption, initial ratings is set to be r=300 Names of vector should correspond with team_name label.
sigma	named vector of rating volatile. In there is no assumption, initial ratings should be sigma=0.5. Names of vector should correspond with team_name label.

tau	The system constant. Which constrains the change in volatility over time. Reasonable choices are between 0.3 and 1.2 (‘default = 0.5’), though the system should be tested to decide which value results in greatest predictive accuracy. Smaller values of ‘tau’ prevent the volatility measures from changing by large amounts, which in turn prevent enormous changes in ratings based on very improbable results. If the application of Glicko-2 is expected to involve extremely improbable collections of game outcomes, then ‘tau’ should be set to a small value, even as small as, say, ‘tau= 0’.2.
weight	name of column in ‘data’ containing weights. Weights increasing or decreasing update change. Higher weight increasing impact of corresponding event.
kappa	controls ‘rd’ shrinkage not to be greater than ‘rd*(1-kappa)’. ‘kappa=1’ means that ‘rd’ will not be decreased.
idlab	name of column in ‘data’ containing date. Doesn’t affect estimation process. If specified, charts displays estimates changes in time instead of by event ‘id’
init_r	initial values for ‘r’ if not provided. Default = 1500
init_rd	initial values for ‘r’ if not provided. Default = 350
pb	logical, if TRUE progress bar will appear in console. Default = FALSE

Value

A "rating" object is returned

- final_r named vector containing players ratings.
- final_rd named vector containing players ratings deviations.
- final_sigma named vector containing players ratings volatiles.
- r data.frame with evolution of the ratings and ratings deviations estimated at each event.
- pairs pairwise combinations of players in analysed events with prior probability and result of a challenge.
- class of the object
- method type of algorithm used
- formula modelled formula

Examples

```
# Example from Glickman
data <- data.frame( name = c( "A", "B", "C", "D" ),
                   rank = c( 3, 4, 1, 2 ))
glicko2 <- glicko2_run( rank ~ name, data )
```

glicko_run	<i>Glicko rating algorithm</i>
------------	--------------------------------

Description

Glicko rating algorithm Wrapper around 'glicko' update algorithm. Wrapper allows user to simplify calculation providing only data and initial parameters assumptions

Usage

```
glicko_run(formula, data, r, rd, sigma, weight, kappa = 0.5, idlab,
           init_r = 1500, init_rd = 350, pb = FALSE)
```

Arguments

formula	formula specifying model. Glicko algorithm allows only player ranking parameter and should be specified by following manner: 'rank id ~ name'. Names in formula are unrestricted, but model structure remains the same: <ul style="list-style-type: none"> • rank player position in event. • id event identifier in which pairwise comparison is assessed. • name of player.
data	data.frame which contains columns specified in formula, and optionally columns defined by 'sigma', 'weight' or 'date'.
r	named vector of initial players ratings estimates. In there is no assumption, initial ratings are set be r=1500. Names of vector should correspond with 'name' in formula.
rd	named vector of initial rating deviation estimates. In there is no assumption, initial ratings are set be r=300 Names of vector should correspond with 'name' in formula.
sigma	name of column in 'data' containing rating volatility. Rating volatility is a value which multiplies prior 'rd'. If 'sigma > 0' then prior 'rd' increases, making estimate of 'r' more uncertain.
weight	name of column in 'data' containing weights. Weights increasing or decreasing update change. Higher weight increasing impact of corresponding event.
kappa	controls 'rd' shrinkage not to be greater than 'rd*(1-kappa)'. 'kappa=1' means that 'rd' will not be decreased.
idlab	name of column in 'data' containing date. Doesn't affect estimation process. If specified, charts displays estimates changes in time instead of by observation 'id'.
init_r	initial values for 'r' if not provided. Default = 1500
init_rd	initial values for 'r' if not provided. Default = 350
pb	logical, if TRUE progress bar will appear in console. Default = FALSE

Value

A "rating" object is returned:

- `final_r` named vector containing players ratings.
- `final_rd` named vector containing players ratings deviations.
- `r` data.frame with evolution of the ratings and ratings deviations estimated at each event.
- `pairs` pairwise combinations of players in analysed events with prior probability and result of a challenge.
- class of the object.
- method type of algorithm used.
- `formula` modelled formula.

Examples

```
# Example from Glickman
data <- data.frame( name = c( "A", "B", "C", "D" ),
                   rank = c( 3, 4, 1, 2 ))
glicko <- glicko_run( rank ~ name, data )
```

gpheats

Heat results of Speedway Grand-Prix

Description

Actual dataset containing heats results of all Speedway Grand-Prix tournaments gpheats.

Format

A data frame with >19000 rows and 11 variables:

id event identifier
season year of Grand-Prix, 1995-now
date date of tournament
round round in season
name Tournament name
heat heat number, 1-23
field number of gate, 1-4
rider rider name, string
points points gained, integer
position position at finish line, string
rank rank at finish line, integer

Source

internal

gpsquads	<i>Turnament results of Speedway Grand-Prix</i>
----------	---

Description

Actual dataset containing turnament results of all Speedway Grand-Prix events gpsquads

Format

A data frame with >4000 rows and 9 variables:

id event identifier
season year of Grand-Prix, 1995-now
date date of turnament
place stadium of event
round round in season
name Turnament name
rider rider names, 1-6
points points gained, integer
classification classification after an event

Source

internal

plot.rating	<i>Plot rating object</i>
-------------	---------------------------

Description

Plot rating object

Usage

```
## S3 method for class 'rating'
plot(x, n = 10, players, ...)
```

Arguments

x	of class rating
n	number of players to be plotted
players	optional vector with names of the players (coefficients) to plot their evolution in time.
...	optional arguments

predict.rating *Predict rating model*

Description

Predict rating model

Usage

```
## S3 method for class 'rating'  
predict(object, newdata, ...)
```

Arguments

object	of class rating
newdata	data.frame with data to predict
...	optional arguments

Value

probabilities of winning challenge by player over his opponent in all provided events.

Examples

```
glicko <- glicko_run(rank|id~rider, gpheats[1:16,])  
predict(glicko, gpheats[17:20,])
```

summary.rating *Summarizing rating objects*

Description

Summarizing rating objects Summary for object of class 'rating'

Usage

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

Arguments

object	of class rating
...	optional arguments

Value

List with following elements

- formula modelled formula.
- method type of algorithm used.
- Overall Accuracy named vector containing players ratings.
- r data.frame summarized players ratings and model winning probabilities. Probabilities are returned only in models with one variable (ratings)
 - name of a player
 - r players ratings
 - rd players ratings deviation
 - `Model probability` mean predicted probability of winning the challenge by the player.
 - `True probability` mean observed probability of winning the challenge by the player.
 - `Accuracy` Accuracy of prediction.
 - `pairings` number of pairwise occurrences.

Examples

```
model <- glicko_run(rank|id~rider, gpheats[1:100,])
summary(model)
```

Index

`bbt_run`, [2](#)

`dbl_run`, [4](#)

`glicko2_run`, [5](#)

`glicko_run`, [7](#)

`gpheats`, [8](#)

`gpsquads`, [9](#)

`plot.rating`, [9](#)

`predict.rating`, [10](#)

`summary.rating`, [10](#)