Package: mtrank (via r-universe)

February 27, 2025

Title Ranking using Probabilistic Models and Treatment Choice Criteria

Version 0.1-1

Date 2025-02-26

Depends R (>= 4.0.0), meta (>= 8.0-2), netmeta (>= 3.0-2)

Imports PlackettLuce, dplyr, magrittr

Suggests rmarkdown, knitr

Maintainer Theodoros Evrenoglou

<theodoros.evrenoglou@uniklinik-freiburg.de>

URL https://github.com/TEvrenoglou/mtrank

Description Implementation of a novel frequentist approach to produce clinically relevant treatment hierarchies in network meta-analysis. The method is based on treatment choice criteria (TCC) and probabilistic ranking models, as described by Evrenoglou et al. (2024) <DOI:10.48550/arXiv.2406.10612>. The TCC are defined using a rule based on the minimal clinically important difference. Using the defined TCC, the study-level data (i.e., treatment effects and standard errors) are first transformed into a preference format, indicating either a treatment preference (e.g., treatment A > treatment B) or a tie (treatment A = treatment B). The preference data are then synthesized using a probabilistic ranking model, which estimates the latent ability parameter of each treatment and produces the final treatment hierarchy. This parameter represents each treatment's ability to outperform all the other competing treatments in the network. Consequently, larger ability estimates indicate higher positions in the ranking list.

License GPL (>= 2)

Encoding UTF-8

RoxygenNote 7.3.2

VignetteBuilder knitr

NeedsCompilation no

Config/pak/sysreqs cmake libglpk-dev libgmp3-dev make libicu-dev libxml2-dev libmpfr-dev libx11-dev

Repository https://tevrenoglou.r-universe.dev

RemoteUrl https://github.com/tevrenoglou/mtrank

RemoteRef HEAD

RemoteSha 0042ee9678586c24b95a66d2e8451c435adbf9a8

Contents

`mtrank-package` *mtrank: Brief overview*

Description

R package **mtrank** enables the production of clinically relevant treatment hierarchies in network meta-analysis using a novel frequentist approach based on treatment choice criteria (TCC) and probabilistic ranking models, as described by Evrenoglou et al. (2024). The TCC are defined using a rule based on the minimal clinically important difference. Using the defined TCC, the study-level data (i.e., treatment effects and standard errors) are first transformed into a preference format, indicating either a treatment preference (e.g., treatment A > treatment B) or a tie (treatment A = treatment B). The preference data are then synthesized using a probabilistic ranking model, which estimates the latent ability parameter of each treatment and produces the final treatment hierarchy. This parameter represents each treatment's ability to outperform all the other competing treatments in the network. Consequently, larger ability estimates indicate higher positions in the ranking list.

Details

The R package **mtrank** provides the following functions:

- Function `tcc` defines the TCC and transforms the study-specific relative treatment effects into a preference format.
- Function `mtrank` synthesizes the output of the `tcc` function and estimates the final treatment ability.

- Forest plots are created either for the results of the TCC (`forest.tcc`) or the final ability estimates (`forest.mtrank`).
- Function `paired_pref` uses the ability estimates obtained from `mtrank` to calculate pairwise probabilities that any treatment 'A' can be better, equal, or worse than any other treatment 'B' in the network.

Type `help(package = "mtrank")` for a listing of R functions available in **mtrank**.

Type `citation("mtrank")` on how to cite **mtrank** in publications.

To report problems and bugs, please send an email to Theodoros Evrenoglou <theodoros.evrenoglou@uniklinik-freiburg.de>.

The development version of **mtrank** is available on GitHub https://github.com/TEvrenoglou/mtrank.

Author(s)

Theodoros Evrenoglou <theodoros.evrenoglou@uniklinik-freiburg.de>, Guido Schwarzer <guido.schwarzer@uniklinik-freiburg.de>

References

Evrenoglou T, Nikolakopoulou A, Schwarzer G, Rücker G, Chaimani A (2024): Producing treatment hierarchies in network meta-analysis using probabilistic models and treatment-choice criteria. https://arxiv.org/abs/2406.10612

See Also

Useful links:

- https://github.com/TEvrenoglou/mtrank

---

antidepressants *Network meta-analysis for major depressive disorder*

Description

Network meta-analysis comparing antidepressants in patients with major depressive disorder.

Format

A data frame with the following columns:

studyid	study id
drug_name	antidepressant name
ntotal	number of randomized patients in treatment arm
responders	number of responders

Source

Cipriani A, Furukawa T, Salanti G, Chaimani A, et al. (2018): Comparative efficacy and acceptability of 21 antidepressant drugs for the acute treatment of adults with major depressive disorder: a systematic review and network meta-analysis *Lancet*, **391**, 1357–1366

See Also

mtrank, tcc

Examples

```
data(antidepressants)
#
ranks <- tcc(treat = drug_name, studlab = studyid,
  event = responders, n = ntotal, data = antidepressants,
  mcid = 1.25, sm = "OR", small.values = "undesirable")
# Visualize treatment choice criteron for treatment "escitalopram"
forest(ranks, treat = "escitalopram")
# Fit the model
fit <- mtrank(ranks)
# Visualize the results
forest(fit)
# Calculate pairwise probabilities
paired_pref(fit, type = "better",
  treat1 = "bupropion", treat2 = "escitalopram")



# Same results using pairwise object as input to tcc()
# (and running a network meta-analysis)
#
pw <- pairwise(treat = drug_name, studlab = studyid,
  event = responders, n = ntotal, data = antidepressants,
  sm = "OR")
#
netmeta(pw)
#
ranks2 <- tcc(pw, mcid = 1.25, small.values = "undesirable")
#
fit2 <- mtrank(ranks2)
#
paired_pref(fit2, type = "better",
  treat1 = "bupropion", treat2 = "escitalopram")
```

---

diabetes *Network meta-analysis studying the incidence of diabetes*

Description

Network meta-analysis comparing six antihypertensive drugs against the incidence of diabetes.

Format

A data frame with the following columns:

study	study label
id	study id
t	treatment label
r	number of events
n	group sample size
rob	risk of bias assessment

Source

Elliott W, Meyer P (2007): Incident diabetes in clinical trials of antihypertensive drugs: a network meta-analysis *Lancet*, **369**

See Also

mtrank, tcc

Examples

```
data(diabetes)
#
ranks <- tcc(treat = t, studlab = study, event = r, n = n, data = diabetes,
  mcid = 1.20, sm = "OR", small.values = "desirable")
#
forest(ranks, treat = "ARB")
```

---

forest.mtrank *Forest plot of ability estimates produced with* mtrank

---

Description

This function produces a forest plot that visualizes the ability estimates calculated with mtrank.

Usage

```
## S3 method for class 'mtrank'
forest(
  x,
  sorting = "ability",
  backtransf = FALSE,
  xlab = "",
  leftcols = "studlab",
```

```
  leftlabs = "Treatment",
  rightcols = c("effect", "ci"),
  rightlabs = c(paste0(if (!backtransf) "log-", "Abilities"), NA),
  label.left = "Favors average treatment",
  label.right = "Favors treatment",
  header.line = TRUE,
  ...
)
```

Arguments

x	An object of class mtrank.
sorting	An argument specifying the criterion to sort the ability estimates in the forest plot (see Details).
backtransf	A logical argument specifying whether to show log-ability estimates (FALSE, default) or ability estimates on the natural scale (TRUE).
xlab	A label for the x-axis.
leftcols	A character vector specifying columns to be printed on the left side of the forest plot (see forest.meta).
leftlabs	A character vector specifying labels for columns on left side of the forest plot.
rightcols	A character vector specifying columns to be printed on the right side of the forest plot (see forest.meta).
rightlabs	A character vector specifying labels for columns on right side of the forest plot.
label.left	Graph label on left side of null effect.
label.right	Graph label on right side of null effect.
header.line	A logical value indicating whether to print a header line or a character string ("both", "below", "").
...	Additional arguments (passed on to forest.meta).

Details

The function produces a forest plot and visualizes the ability estimates obtained from mtrank. The order of the estimates in the forest plot (argument sorting) can be one of the following:

- "ability": sort by descending ability estimates (default),
- "se": sort by descending precision, i.e., increasing standard errors,
- "none": use order from data set.

Value

A forest plot is plotted in the active graphics device.

References

Evrenoglou T, Nikolakopoulou A, Schwarzer G, Rücker G, Chaimani A (2024): Producing treatment hierarchies in network meta-analysis using probabilistic models and treatment-choice criteria. https://arxiv.org/abs/2406.10612

Examples

```
data(antidepressants)
#
ranks <- tcc(treat = drug_name, studlab = studyid,
  event = responders, n = ntotal, data = antidepressants,
  mcid = 1.25, sm = "OR", small.values = "undesirable")
#
fit <- mtrank(ranks)

forest(fit, treat = "escitalopram")
```

`forest.tcc` *Forest plot showing study-specific preferences or ties according to treatment choice criterion*

Description

This function produces a forest plot for all (or selected) study specific comparisons and visualizes the treatment preference or ties which are defined from the treatment choice criterion in `tcc`.

Usage

```
## S3 method for class 'tcc'
forest(
  x,
  treat = NULL,
  backtransf = FALSE,
  leftcols = "studlab",
  leftlabs = NULL,
  rightcols = c("effect", "ci"),
  lty.equi = gs("lty.equi"),
  col.equi = gs("col.equi"),
  fill.equi = gs("fill.equi"),
  fill.lower.equi = fill.equi,
  fill.upper.equi = rev(fill.equi),
  header.line = TRUE,
  col.subgroup = "black",
  ...
)
```

Arguments

`x`	An object of class `tcc`.
`treat`	A treatment of interest. If specified it returns a forest plot for all study specific effects related to `treat`. If NULL (default), it generates a forest plot for all study-specific effects in the network.

backtransf	A logical indicating whether results should be back transformed. If backtransf = TRUE (default), results for sm = "OR" are printed as odds ratios rather than log odds ratios, for example.
leftcols	A character vector specifying columns to be printed on the left side of the forest plot (see forest.meta).
leftlabs	A character vector specifying labels for columns on left side of the forest plot.
rightcols	A character vector specifying columns to be printed on the right side of the forest plot (see forest.meta).
lty.equi	Line type (limits of equivalence).
col.equi	Line colour (limits of equivalence).
fill.equi	Colour(s) for area between limits of equivalence or more general limits.
fill.lower.equi	Colour of area between lower limit(s) and reference value. Can be equal to the number of lower limits or the number of limits plus 1 (in this case the the region between minimum and smallest limit is also filled).
fill.upper.equi	Colour of area between reference value and upper limit(s). Can be equal to the number of upper limits or the number of limits plus 1 (in this case the region between largest limit and maximum is also filled).
header.line	A logical value indicating whether to print a header line or a character string ("both", "below", "").
col.subgroup	The colour to print information on subgroups, i.e., pairwise comparisons.
...	Additional arguments (passed on to forest.meta).

Details

This function produces forest plots for the study specific treatment effects in the network. The color indicates whether treatment effects show a preference (red color) or tie (black color). Additionally, the respective range of equivalence defined at the function tcc is visualized for the forest plot.

Argument treat is optional. By default (treat = NULL), all study-specific treatment effects in the network are shown. If specified, only study-specific treatment effects related to the specified treat are shown which is useful in busy networks with many direct comparisons.

Value

A forest plot is plotted in the active graphics device.

References

Evrenoglou T, Nikolakopoulou A, Schwarzer G, Rücker G, Chaimani A (2024): Producing treatment hierarchies in network meta-analysis using probabilistic models and treatment-choice criteria. https://arxiv.org/abs/2406.10612

Examples

```
data(diabetes)
#
ranks <- tcc(treat = t, studlab = study, event = r, n = n, data = diabetes,
  mcid = 1.20, sm = "OR", small.values = "desirable")
#
forest(ranks)
forest(ranks, treat = "ARB")
```

`mtrank` *Estimate the treatment hierarchy in network meta-analysis using a probabilistic ranking model*

Description

This function fits the Bradley-Terry ranking model and produces a treatment hierarchy based on the method described by Evrenoglou et al. (2024) for network meta-analysis.

Usage

```
mtrank(x, reference.group = NULL, level = x$level)

## S3 method for class 'mtrank'
print(
  x,
  sorting = "ability",
  backtransf = FALSE,
  digits = gs("digits"),
  digits.prop = gs("digits.prop"),
  ...
)
```

Arguments

`x`	An object of class `tcc` or `mtrank` (print function).
`reference.group`	An argument specifying the reference group. If set to NULL (default), ability estimates of all treatments will be calculated. If some treatment is set as the reference group, relative abilities of all treatments versus the specified reference treatment will be calculated.
`level`	The level used to calculate confidence intervals for ability estimates.
`sorting`	An argument specifying the criterion to sort the ability estimates in the printout (see Details).
`backtransf`	A logical argument specifying whether to show log-ability estimates (`FALSE`, default) or on the natural scale (`TRUE`).

digits	Minimal number of significant digits for ability estimates, see print.default.
digits.prop	Minimal number of significant digits for proportions, see print.default.
...	Additional arguments (passed on to prmatrix).

Details

This function is used to fit a Bradley-Terry model to the paired-preference data generated from the treatment choice criterion constructed by the tcc function. This function estimates the ability of each treatment in the network and the respective standard errors and confidence intervals using the maximum likelihood approach. To retain identifiability, the maximization of the log-likelihood takes place subject to the constrain that the ability estimates sum to 1. Then, the maximum likelihood estimates (MLEs) are calculated iteratively. Note that the final estimates of the ability parameters are not necessarily needed to sum to 1 as after the first iteration of the algorithm the ability estimates are not normalized. However, by normalizing the final ability estimates to sum up to 1 these can be interpreted as "the probability that each treatment is having the highest ability".

Finally, a parameter "v" controlling the prevalence of ties in the network is also estimated. Although the estimated values of this parameter do not have a direct interpretation they are useful for estimating pairwise probabilities (see paired_pref).

If argument reference.group is not NULL, a reference treatment group is specified. Mathematically, this means that the maximization problem is now identifiable, subject to the condition that the ability of this treatment is 0. Then, the resulting MLEs are the relative abilities of all treatments in the network versus the specified reference treatment group. Note that the estimates of the parameter "v" and the normalized probabilities do not depend on the value for argument reference.group.

Value

- A data frame containing the resulting log-ability estimates, their standard errors and their confidence intervals.
- The estimate of the tie prevalence parameter v.
- The normalized ability estimates for each treatment.

References

Evrenoglou T, Nikolakopoulou A, Schwarzer G, Rücker G, Chaimani A (2024): Producing treatment hierarchies in network meta-analysis using probabilistic models and treatment-choice criteria. https://arxiv.org/abs/2406.10612

Examples

```
data(antidepressants)

ranks <- tcc(treat = drug_name, studlab = studyid,
  event = responders, n = ntotal, data = antidepressants,
  mcid = 1.25, sm = "OR", small.values = "undesirable")
#
fit1 <- mtrank(ranks)
#
# Print log-ability estimates
```

```
fit1
#
# Print ability estimates
print(fit1, backtransf = TRUE)
# Visualize results
 forest(fit1)

# Repeat using a 'pairwise' object
pw <- pairwise(treat = drug_name, studlab = studyid,
  event = responders, n = ntotal, data = antidepressants,
  sm = "OR")

ranks2 <- tcc(pw, mcid = 1.25, small.values = "undesirable")
#
fit2 <- mtrank(ranks2)

# Print log-ability estimates
fit2
# Print ability estimates
print(fit2, backtransf = TRUE)
# Visualize results
 forest(fit2)
```

`paired_pref` *Calculate pairwise probabilities for* `mtrank` *object*

Description

This function uses the estimates of ability and tie prevalence parameters from a `mtrank` object and calculates pairwise probabilities about the preference or the tie between two treatments based on equations (7) and (8) in Evrenoglou et al. (2024).

Usage

```
paired_pref(x, treat1, treat2, type)

## S3 method for class 'paired_pref'
print(x, type = attr(x, "type"), digits = 4, ...)
```

Arguments

`x`	An object of class `mtrank`.
`treat1`	The first treatment considered in the treatment comparison.
`treat2`	The second treatment considered in the treatment comparison.
`type`	A character vector specifying the probability of interest. Either "better", "tie", "worse", or "all" (can be abbreviated).
`digits`	Minimal number of significant digits for proportions, see `print.default`.
`...`	Additional arguments (passed on to `prmatrix`).

Details

Pairwise probabilities between any two treatments in the network can be calculated using the ability estimates obtained from mtrank and equations (7) and (8) in Evrenoglou et al. (2024). The probabilities are calculated in the direction treat1 vs treat2. The available probability types are

- "better": probability that treat1 is better than treat2,
- "tie": probability that treat1 is equal to treat2,
- "worse": probability that treat1 is worse than treat2,
- "all": all three probabilities.

Please note that all the arguments of this function are mandatory.

Value

The probability (or probabilities) of interest for the comparison treat1 vs treat2 based on the argument type.

References

Evrenoglou T, Nikolakopoulou A, Schwarzer G, Rücker G, Chaimani A (2024): Producing treatment hierarchies in network meta-analysis using probabilistic models and treatment-choice criteria. https://arxiv.org/abs/2406.10612

Examples

```
data(antidepressants)
#
ranks <- tcc(treat = drug_name, studlab = studyid,
  event = responders, n = ntotal, data = antidepressants,
  mcid = 1.25, sm = "OR", small.values = "undesirable")
#
fit <- mtrank(ranks)
#
paired_pref(fit, type = c("better", "worse"),
  treat1 = "bupropion", treat2 = "escitalopram")
#
paired_pref(fit, type = c("better", "worse"),
  treat1 = "escitalopram", treat2 = "bupropion")
#
paired_pref(fit, type = "all",
  treat1 = c("bupropion", "escitalopram"),
  treat2 = c("escitalopram", "bupropion"))
```

pp2long *Auxiliary function to transform data from paired-preference to long-arm format*

Description

Auxiliary function to transform data from paired-preference to long-arm format

Usage

```
pp2long(x)
```

Arguments

x	An object of class "ppdata" (part of tcc object).

Value

Data set in long-arm format that can be used as input to rankings.

Author(s)

Guido Schwarzer <guido.schwarzer@uniklinik-freiburg.de>

See Also

tcc, rankings

Examples

```
data(diabetes)
#
ranks <- tcc(treat = t, studlab = study, event = r, n = n, data = diabetes,
  mcid = 1.20, sm = "OR", small.values = "desirable")
#
pdat <- ranks$ppdata
#
ldat <- pp2long(pdat)
head(ldat)

library("PlackettLuce")
ungrouped.preferences <-
  rankings(ldat, id = "id", item = "treat", rank = "rank")
grouped.preferences <-
  as.rankings(ungrouped.preferences,
              index = as.numeric(as.factor(pdat$studlab)))
#
fit <- PlackettLuce(grouped.preferences)
#
```

```
coef(summary(fit, ref = ranks$reference.group))[, 1]
# Results stored in mtrank()
mtrank(ranks)$estimates$log_ability
```

---

tcc *Transform meta-analysis data from long or wide arm-based format into paired-preference format*

Description

This function transforms data that are given in wide or long arm-based format (e.g. input format for WinBUGS or JAGS) to a paired-preference format needed as input to `mtrank`. The function can transform data with binary and continuous arm-based to preference-based format.

Usage

```
tcc(
  treat,
  event,
  n,
  mean,
  sd,
  data = NULL,
  studlab,
  mcid = NULL,
  lower.equi = NULL,
  upper.equi = NULL,
  small.values = gs("small.values"),
  relax = FALSE,
  level = 0.95,
  sm,
  keepdata = gs("keepdata"),
  ...
)

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

Arguments

treat	Either a `pairwise` object, or a list or vector with treatment information for individual treatment arms (see Details).
event	A list or vector with information on number of events for individual treatment arms (see Details).
n	A list or vector with information on number of observations for individual treatment arms (see Details).

mean	A list vector with estimated means for individual treatment arms (see Details).
sd	A list or vector with information on the standard deviation for individual treatment arms (see Details).
data	A data frame containing the study information.
studlab	A vector with study labels.
mcid	A numeric vector specifying the minimal clinically important value (see Details).
lower.equi	A numeric value specifying the lower limit of the range of equivalence (see Details).
upper.equi	A numeric value specifying the upper limit of the range of equivalence (see Details).
small.values	A character string specifying whether small treatment effects indicate a beneficial ("desirable") or harmful ("undesirable") effect.
relax	A logical optional argument. If TRUE it 'relaxes' the tcc to only consider the bounds of ROE when specifying 'wins' and ties. The default FALSE uses the criterion described by Evrenoglou et al. (2024) and considers also the statistical significance on top of the ROE bounds (see Details).
level	The level used to calculate confidence intervals for log-abilities.
sm	The effect measure of interest (see Details).
keepdata	A logical indicating whether original data should be kept in tcc object.
...	Additional arguments (passed on to pairwise).
x	An object of class tcc.

Details

R function mtrank expects data in a **paired-preference** format, where for each study-specific pairwise comparison in the network a treatment preference or tie is indicated. For example, for the study-specific comparison between treatments A and B the potential outcomes are:

- $A > B$
- $A < B$
- $A = B$

The data transformation takes place based on the study-specific treatment effects and the treatment choice criterion. R function pairwise is called internally to calculate the study-specific treatment effect estimates and standard errors. This ensures that data given in either 'long' or 'wide' arm-based format will be suitably used to calculate the study-specific treatment effect estimates and standard errors while ensuring that a network of multi-arm studies gets an equivalent representation as a network of two-arm studies. It is also possible to provide a pairwise as the main input. In this case, inputs for the arguments event, n, mean, sd, data, studlab, or keepdata are ignored.

This function implements treatment choice criteria based on the method by Evrenoglou et al. (2024). Namely, a range of equivalence (ROE) can be specified by

- argument mcid. Then the limits of the ROE will be defined based on the values (i) mcid, 1/mcid for ratio measures and (ii) mcid and -mcid for difference measures.

- arguments `lower.equi` and `upper.equi`. These arguments allow the users to define their own limits of the ROE, given the restriction that the lower limit will always be smaller than the upper limit.

Note that when the argument `mcid` is specified, the arguments `lower.equi` and `upper.equi` are ignored. Either only the `mcid` or both of the `lower.equi` and `upper.equi` must be specified for the proper definition of the ROE.

After setting the ROE, each study-specific treatment effect will be categorised as a treatment preference or a tie. The argument `relax` controls the amount of conservatism of the treatment choice criterion. If set to `FALSE` (default), the treatment choice criterion is equivalent to the one described by Evrenoglou et al. (2024). In this case, study-specific treatment effects need to be both statistically and clinically significant to indicate a treatment preference. If set to `TRUE`, the criterion is relaxed and the study-specific treatment effects need to be only clinically significant to indicate a treatment preference.

This function can transform data with binary and continuous outcomes. Depending on the outcome, the following arguments are mandatory:

- treat, event, n (for binary outcomes);
- treat, n, mean, sd (for continuous outcomes).

Finally, the argument `sm` is used to define the effect measure of interest for transforming the data into paired-preference format; see `metabin` and `metacont` for a list of available effect measures.

Value

- The initial data in a paired-preference format.
- The correspondence between the initial study names (passed in the argument studlab) and the index name of the paired-preference format data.

References

Evrenoglou T, Nikolakopoulou A, Schwarzer G, Rücker G, Chaimani A (2024): Producing treatment hierarchies in network meta-analysis using probabilistic models and treatment-choice criteria. https://arxiv.org/abs/2406.10612

Examples

```
data(diabetes)
#
ranks <- tcc(treat = t, studlab = study, event = r, n = n, data = diabetes,
  mcid = 1.20, sm = "OR", small.values = "desirable")
#
forest(ranks, treat = "ARB")
```

Index