

Package ‘BiasedMatch’

October 21, 2026

Type Package

Title Biased Matching for Two-Urn Sampling Problems

Version 0.1.0

Description Tools for the two-urn biased sampling problems of Puza and Bonfrer (2018) <[doi:10.1080/03610926.2017.1300282](https://doi.org/10.1080/03610926.2017.1300282)>. Two overlapping sets of items are sampled without replacement, with items common to both sets given a selection weight favouring their inclusion. The package computes the probability mass function, mean and variance of the number of matching items using the univariate and bivariate Wallenius' non-central hypergeometric distributions, performs maximum likelihood and Bayesian inference for the bias parameter, and provides Metropolis-Hastings algorithms for hierarchical extensions.

Depends R (>= 3.5.0)

Imports BiasedUrn, mvtnorm, nortest, EnvStats, coda, stats, utils

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

License MIT + file LICENSE

Encoding UTF-8

URL <https://github.com/andrebonfrer/BiasedMatch>

BugReports <https://github.com/andrebonfrer/BiasedMatch/issues>

Config/testthat/edition 3

VignetteBuilder knitr

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Andre Bonfrer [aut, cre] (ORCID:
<<https://orcid.org/0000-0002-3061-7103>>),
Borek Puza [aut]

Maintainer Andre Bonfrer <andre.bonfrer@deakin.edu.au>

Repository CRAN

Date/Publication 2026-10-21 09:40:16 UTC

Contents

DISTM	2
estprop	3
inferW	4
likW	6
MHALG	7
PROBM	9
restable	10
RMMHALG	11
SMRMMHALG	12
Vmf	14
Index	16

DISTM	<i>Distribution, mean and variance of the number of matches</i>
-------	---

Description

Computes the complete probability distribution of the number of matches together with its mean and variance, for the biased two-urn sampling model.

Usage

DISTM(N1 = 37, N2 = 45, N = 16, m1 = 12, m2 = 8, w = 2)

Arguments

N1	Integer; number of distinct items in urn 1.
N2	Integer; number of distinct items in urn 2.
N	Integer; number of items common to both urns.
m1	Integer; sample size drawn from urn 1.
m2	Integer; sample size drawn from urn 2.
w	Numeric (> 0); weight of a common item relative to a non-common item. w = 1 gives unbiased simple random sampling without replacement.

Value

A list with components:

- mvec** Integer vector of possible match counts.
- fmvec** Probability masses corresponding to mvec.
- sumfmvec** Sum of fmvec; a numerical check that should equal 1.
- Em** Expected number of matches.
- Vm** Variance of the number of matches.
- SDm** Standard deviation of the number of matches.

References

Puza, B. and Bonfrer, A. (2018). A series of two-urn biased sampling problems. *Communications in Statistics - Theory and Methods*, 47(1), 80-91. doi:[10.1080/03610926.2017.1300282](https://doi.org/10.1080/03610926.2017.1300282)

See Also

[PROBM\(\)](#) for a single probability, [Vmf\(\)](#) for the closed-form unbiased variance.

Examples

```
# Full distribution for Example 1 of the paper (w = 2)
d <- DISTM(N1 = 37, N2 = 45, N = 16, m1 = 12, m2 = 8, w = 2)
d$sumfmvec # ~ 1
d$Em # expected number of matches
```

 estprop

Convergence and serial-correlation diagnostics for MH draws

Description

Applies a suite of diagnostic tests to the thinned posterior draws returned by [MHALG\(\)](#), [RMMHALG\(\)](#) or [SMRMMHALG\(\)](#): rank von Neumann and AR(1) Yule-Walker serial-correlation tests (**EnvStats**), the Anderson-Darling, Cramer-von Mises and Lilliefors normality tests (**nortest**), and the Geweke convergence diagnostic (**coda**).

Usage

```
estprop(resg, Bn = 500, Jn = 2000, s = 4, step = 10)
```

Arguments

resg	List; output from one of the MH samplers.
Bn	Integer; number of initial iterations to discard as burn-in.
Jn	Integer; number of post-burn-in iterations to use.
s	Integer; starting index for thinning.
step	Integer; thinning step size.

Value

A data frame of p-values with one row per diagnostic test and one column per parameter (μ , λ , and each weight class).

See Also

[MHALG\(\)](#), [restable\(\)](#).

Examples

```
set.seed(7204)
res <- MHALG(
  B = 10, J = 200,
  N1vec = 37, N2vec = 45, N0vec = 16,
  M1vec = 12, M2vec = 8, mvec = 3, Cvec = 1, del = 0.1
)
estprop(res, Bn = 10, Jn = 150, s = 1, step = 2)
```

inferW

Inference on the bias parameter from an observed number of matches

Description

"Inverts" the biased sampling model of `PROBM()`: given the fixed urn constants and a single observed number of matches m , it returns the maximum likelihood estimate and posterior mode of the bias parameter w , and a p-value for testing no bias against a positive bias. This implements the inferential problems (Problems 1-3) of Puza and Bonfrer (2018).

Usage

```
inferW(
  m,
  N1,
  N2,
  N,
  m1,
  m2,
  prior = c("jeffreys", "flat"),
  wmax = 200,
  alternative = c("greater", "less")
)
```

Arguments

<code>m</code>	Integer; number of matches for which the probability is required.
<code>N1</code>	Integer; number of distinct items in urn 1.
<code>N2</code>	Integer; number of distinct items in urn 2.
<code>N</code>	Integer; number of items common to both urns.
<code>m1</code>	Integer; sample size drawn from urn 1.
<code>m2</code>	Integer; sample size drawn from urn 2.
<code>prior</code>	Character; the prior on w used for the posterior mode. "jeffreys" (the default) uses $f(w)$ proportional to $1/w$, as in the paper; "flat" uses a uniform prior, so the posterior mode equals the MLE.

<code>wmax</code>	Numeric; upper bound of the search region for w . If the likelihood is still increasing at <code>wmax</code> , the MLE is reported as infinite.
<code>alternative</code>	Character; the alternative hypothesis for the p-value. "greater" (default) tests $w = 1$ against $w > 1$ (bias towards common items); "less" tests against $w < 1$.

Details

The likelihood is $L(w) = \text{PROBM}(m, \dots, w)$ and the posterior kernel under the Jeffreys-type prior is $L(w) / w$. Both are maximised numerically on $(0, wmax]$. As noted in the paper, when the observed m is large enough the likelihood is strictly increasing and the MLE is infinite; this is detected and reported as `Inf`, while the posterior mode remains finite. The p-value is the null-hypothesis ($w = 1$) tail probability of the number of matches, obtained from `DISTM()`.

Value

A list with components:

MLE Maximum likelihood estimate of w (possibly `Inf`).

posterior_mode Posterior mode of w under prior.

p_value P-value for testing $w = 1$, computed from the null ($w = 1$) distribution of the number of matches.

alternative The alternative hypothesis used.

m The observed number of matches.

References

Puza, B. and Bonfrer, A. (2018). A series of two-urn biased sampling problems. *Communications in Statistics - Theory and Methods*, 47(1), 80-91. doi:[10.1080/03610926.2017.1300282](https://doi.org/10.1080/03610926.2017.1300282)

See Also

`likW()` for the full likelihood and posterior curves.

Examples

```
# Problem 2 of the paper (gender bias in committee selection, B = 2).
# Small urns, so this runs quickly.
inferW(m = 2, N1 = 52, N2 = 52, N = 22, m1 = 9, m2 = 6)
```

```
# Problem 3 of the paper (masked retailer promotion data). The larger urns
# make the likelihood search over w slower to run.
inferW(m = 13, N1 = 292, N2 = 350, N = 192, m1 = 90, m2 = 59)
```

likW

*Likelihood and posterior curves for the bias parameter***Description**

Evaluates, over a grid of weight values, the likelihood of the bias parameter w given an observed number of matches m , together with the kernel of the posterior density under the prior $f(w)$ proportional to $1/w$. Both curves are returned standardised to a maximum of 1, matching the figures in Puza and Bonfrer (2018).

Usage

```
likW(m, N1, N2, N, m1, m2, wgrid = seq(0.05, 6, by = 0.05))
```

Arguments

<code>m</code>	Integer; number of matches for which the probability is required.
<code>N1</code>	Integer; number of distinct items in urn 1.
<code>N2</code>	Integer; number of distinct items in urn 2.
<code>N</code>	Integer; number of items common to both urns.
<code>m1</code>	Integer; sample size drawn from urn 1.
<code>m2</code>	Integer; sample size drawn from urn 2.
<code>wgrid</code>	Numeric vector of positive weight values at which to evaluate the curves.

Value

A data frame with columns:

w The weight grid (as supplied).

likelihood Likelihood values standardised to a maximum of 1.

posterior Posterior-kernel values (likelihood / w) standardised to a maximum of 1.

References

Puza, B. and Bonfrer, A. (2018). A series of two-urn biased sampling problems. *Communications in Statistics - Theory and Methods*, 47(1), 80-91. doi:[10.1080/03610926.2017.1300282](https://doi.org/10.1080/03610926.2017.1300282)

See Also

[inferW\(\)](#) for point estimates and the test of $w = 1$.

Examples

```
# Small, fast illustration on a coarse grid
likW(m = 2, N1 = 52, N2 = 52, N = 22, m1 = 9, m2 = 6,
     wgrid = seq(0.5, 4, by = 0.5))

# Reproduce the likelihood/posterior curves of Problem 3 (marketing data).
# Uses a fine grid over the larger urns, so it is slower to run.
curves <- likW(
  m = 13, N1 = 292, N2 = 350, N = 192, m1 = 90, m2 = 59,
  wgrid = seq(0.05, 8, by = 0.05)
)
plot(curves$w, curves$likelihood, type = "l",
     xlab = "w", ylab = "standardised value")
lines(curves$w, curves$posterior, lty = 2)
```

MHALG

*Metropolis-Hastings sampler for a single weight class***Description**

Draws from the posterior distribution of the bias parameter and its hyperparameters for a set of two-urn matching observations that share a common weight. This is the hierarchical extension of the inferential problem of Puza and Bonfrer (2018), and is most useful when several matching experiments are pooled.

Usage

```
MHALG(
  B = 5,
  J = 20,
  N1vec = 150,
  N2vec = 200,
  N0vec = 50,
  M1vec = 40,
  M2vec = 60,
  mvec = 10,
  Cvec = 1,
  eta = 0.001,
  tau = 1000,
  mu0 = 0,
  sig0 = 0.001,
  muinit = 0,
  rinit = 0,
  del = 0.1,
  u = exp(20),
```

```

    v = 1,
    hf = 1
  )

```

Arguments

B	Integer; number of burn-in iterations.
J	Integer; number of sampling iterations retained after burn-in.
N1vec, N2vec	Numeric vectors; N1 and N2 for each observation.
N0vec	Numeric vector; number of common items N for each observation.
M1vec, M2vec	Numeric vectors; sample sizes m1 and m2 for each observation.
mvec	Numeric vector; observed numbers of matches.
Cvec	Integer vector; weight-class index for each observation.
eta, tau	Numeric; shape and rate hyperparameters of the gamma prior on the precision lambda.
mu0, sig0	Numeric; mean and scale hyperparameters of the normal prior on mu.
muinit, rinit	Numeric; starting values for mu and the latent weights.
del	Numeric; random-walk proposal standard deviation(s).
u, v	Numeric; parameters of the internal link function mapping the latent scale to the positive weight w.
hf	Integer; if 0, use the log-scale form of the log full conditional, otherwise the link-function form (the default).

Value

A list with components `muv` (sampled mu), `lamv` (sampled lambda), `rm` (matrix of sampled latent weights, one column per class) and `rar` (acceptance rate per class).

References

Puza, B. and Bonfrer, A. (2018). A series of two-urn biased sampling problems. *Communications in Statistics - Theory and Methods*, 47(1), 80-91. doi:[10.1080/03610926.2017.1300282](https://doi.org/10.1080/03610926.2017.1300282)

See Also

[restable\(\)](#) and [estprop\(\)](#) to summarise and diagnose the output.

Examples

```

# Short illustrative run; use many more iterations for real inference.
set.seed(4172)
res <- MHALG(
  B = 5, J = 20,
  N1vec = 37, N2vec = 45, N0vec = 16,
  M1vec = 12, M2vec = 8, mvec = 3,
  Cvec = 1, del = 0.1
)

```

str(res)	
PROBM	<i>Probability mass for the number of matching items</i>

Description

Computes the probability that exactly m items match across two samples drawn without replacement from two overlapping urns, under the biased selection scheme of Puza and Bonfrer (2018).

Usage

PROBM($m = 3$, $N1 = 37$, $N2 = 45$, $N = 16$, $m1 = 12$, $m2 = 8$, $w = 2$)

Arguments

m	Integer; number of matches for which the probability is required.
$N1$	Integer; number of distinct items in urn 1.
$N2$	Integer; number of distinct items in urn 2.
N	Integer; number of items common to both urns.
$m1$	Integer; sample size drawn from urn 1.
$m2$	Integer; sample size drawn from urn 2.
w	Numeric (> 0); weight of a common item relative to a non-common item. $w = 1$ gives unbiased simple random sampling without replacement.

Details

Urn 1 contains $N1$ distinct items and urn 2 contains $N2$ distinct items, with N items common to both urns. A sample of size $m1$ is drawn from urn 1 and a sample of size $m2$ from urn 2. Each item common to both urns is given selection weight w while every other item has weight 1, so that $w > 1$ makes common items more likely to be selected. A match is an item selected in both samples.

The implementation follows Theorem 2 of Puza and Bonfrer (2018). It sums over $y1$, the number of common items drawn from urn 1, weighting the conditional match probability given $y1$ by the probability of $y1$. The distribution of $y1$ is Wallenius' univariate non-central hypergeometric distribution ([BiasedUrn::dWNCHypergeo\(\)](#)); the conditional distribution of matches given $y1$ uses the bivariate Wallenius' distribution ([BiasedUrn::dMWNCHypergeo\(\)](#)). When $w = 1$ the result reduces to the hypergeometric mixture of Theorem 1.

Value

A single numeric probability.

References

Puza, B. and Bonfrer, A. (2018). A series of two-urn biased sampling problems. *Communications in Statistics - Theory and Methods*, 47(1), 80-91. [doi:10.1080/03610926.2017.1300282](#)

See Also

[DISTM\(\)](#) for the full distribution and [inferW\(\)](#) for inference on w .

Examples

```
# Probability of exactly 3 matches (Example 1 of the paper, w = 2)
PROBM(m = 3, N1 = 37, N2 = 45, N = 16, m1 = 12, m2 = 8, w = 2)
```

restable

Summarise posterior draws from an MH sampler

Description

Produces posterior means, modes and 95% credible intervals for the hyperparameters and weight classes returned by [MHALG\(\)](#), [RMMHALG\(\)](#) or [SMRMMHALG\(\)](#).

Usage

```
restable(
  res,
  Bn = 2,
  Jn = length(res$muv) - Bn,
  rm.idx = 1:dim(res$rm)[2],
  sim = FALSE
)
```

Arguments

<code>res</code>	List; output from one of the MH samplers.
<code>Bn</code>	Integer; number of initial iterations to discard as burn-in.
<code>Jn</code>	Integer; number of retained iterations to summarise.
<code>rm.idx</code>	Integer vector; columns of <code>res\$rm</code> (weight classes) to summarise.
<code>sim</code>	Logical; reserved for comparison against known parameters. Not implemented and must be FALSE.

Value

A data frame with rows Mean, Mode, Q025 and Q975, and columns for μ , λ , σ and each weight class r_1 , r_2 , ...

See Also

[MHALG\(\)](#), [estprop\(\)](#).

Examples

```

set.seed(5093)
res <- MHALG(
  B = 5, J = 40,
  N1vec = 37, N2vec = 45, N0vec = 16,
  M1vec = 12, M2vec = 8, mvec = 3, Cvec = 1, del = 0.1
)
restable(res, Bn = 5)

```

RMMHALG

*Metropolis-Hastings sampler with linear predictors***Description**

Extends [MHALG\(\)](#) so that the latent weight of each class depends on covariates through a design matrix X and regression coefficients with a multivariate normal prior.

Usage

```

RMMHALG(
  B = 5,
  J = 20,
  N1vec = 150,
  N2vec = 200,
  N0vec = 50,
  M1vec = 40,
  M2vec = 60,
  mvec = 10,
  Cvec = 1,
  X,
  B0,
  V0,
  eta = 0.001,
  tau = 1000,
  mu0 = 0,
  sig0 = 0.001,
  muinit = 0,
  rinit = 0,
  del = 0.1
)

```

Arguments

B	Integer; number of burn-in iterations.
J	Integer; number of sampling iterations retained after burn-in.
N1vec, N2vec	Numeric vectors; N1 and N2 for each observation.

N0vec	Numeric vector; number of common items N for each observation.
M1vec, M2vec	Numeric vectors; sample sizes m1 and m2 for each observation.
mvec	Numeric vector; observed numbers of matches.
Cvec	Integer vector; weight-class index for each observation.
X	Numeric matrix; design matrix linking classes to coefficients.
B0	Numeric vector; prior mean of the regression coefficients.
V0	Numeric matrix; prior covariance of the regression coefficients.
eta, tau	Numeric; shape and rate hyperparameters of the gamma prior on the precision lambda.
mu0, sig0	Numeric; mean and scale hyperparameters of the normal prior on mu.
muinit, rinit	Numeric; starting values for mu and the latent weights.
del	Numeric; random-walk proposal standard deviation(s).

Value

A list with components betv (sampled regression coefficients), lamv, rm and rar, as in [MHALG\(\)](#).

References

Puza, B. and Bonfrer, A. (2018). A series of two-urn biased sampling problems. *Communications in Statistics - Theory and Methods*, 47(1), 80-91. doi:[10.1080/03610926.2017.1300282](https://doi.org/10.1080/03610926.2017.1300282)

Examples

```
set.seed(6215)
X <- cbind(1, c(1, 0, 1))
res <- RMMHALG(
  B = 2, J = 5,
  N1vec = c(150, 150, 150), N2vec = c(200, 200, 200),
  N0vec = c(50, 50, 50), M1vec = c(40, 40, 40), M2vec = c(60, 60, 60),
  mvec = c(10, 11, 9), Cvec = c(1, 2, 3),
  X = X, B0 = c(0, 0), V0 = diag(2)
)
str(res)
```

SMRMMHALG

Metropolis-Hastings sampler with shared weights within classes

Description

Variant of [RMMHALG\(\)](#) that infers the weight classes from the design matrix X and constrains observations mapping to the same linear predictor to share a common weight.

Usage

```

SMRMMHALG(
  B = 5,
  J = 20,
  N1vec = 150,
  N2vec = 200,
  N0vec = 50,
  M1vec = 40,
  M2vec = 60,
  mvec = 10,
  X,
  B0,
  V0,
  eta = 0.001,
  tau = 1000,
  mu0 = 0,
  sig0 = 0.001,
  muinit = 0,
  rinit = 0,
  del = 0.1,
  n = length(mvec)
)

```

Arguments

B	Integer; number of burn-in iterations.
J	Integer; number of sampling iterations retained after burn-in.
N1vec, N2vec	Numeric vectors; N1 and N2 for each observation.
N0vec	Numeric vector; number of common items N for each observation.
M1vec, M2vec	Numeric vectors; sample sizes m1 and m2 for each observation.
mvec	Numeric vector; observed numbers of matches.
X	Numeric matrix; design matrix linking classes to coefficients.
B0	Numeric vector; prior mean of the regression coefficients.
V0	Numeric matrix; prior covariance of the regression coefficients.
eta, tau	Numeric; shape and rate hyperparameters of the gamma prior on the precision lambda.
mu0, sig0	Numeric; mean and scale hyperparameters of the normal prior on mu.
muinit, rinit	Numeric; starting values for mu and the latent weights.
del	Numeric; random-walk proposal standard deviation(s).
n	Integer; number of observations (defaults to length(mvec)).

Value

A list with components betv, lamv, rm and rar, as in [RMMHALG\(\)](#).

References

Puza, B. and Bonfrer, A. (2018). A series of two-urn biased sampling problems. *Communications in Statistics - Theory and Methods*, 47(1), 80-91. doi:10.1080/03610926.2017.1300282

Examples

```
set.seed(8391)
X <- cbind(1, c(1, 0, 1))
res <- SMRMMHALG(
  B = 2, J = 5,
  N1vec = c(150, 150, 150), N2vec = c(200, 200, 200),
  N0vec = c(50, 50, 50), M1vec = c(40, 40, 40), M2vec = c(60, 60, 60),
  mvec = c(10, 11, 9),
  X = X, B0 = c(0, 0), V0 = diag(2)
)
str(res)
```

Vmf	<i>Closed-form variance of the number of matches under unbiased sampling</i>
-----	--

Description

Evaluates the closed-form variance of the number of matches when $w = 1$ (simple random sampling without replacement), given in Theorem 1(b) of Puza and Bonfrer (2018). It is provided for comparison with the exact variance returned by [DSTM\(\)](#).

Usage

```
Vmf(N1 = 37, N2 = 45, N = 16, m1 = 12, m2 = 8)
```

Arguments

- N1 Integer; number of distinct items in urn 1.
- N2 Integer; number of distinct items in urn 2.
- N Integer; number of items common to both urns.
- m1 Integer; sample size drawn from urn 1.
- m2 Integer; sample size drawn from urn 2.

Value

A single numeric variance.

References

Puza, B. and Bonfrer, A. (2018). A series of two-urn biased sampling problems. *Communications in Statistics - Theory and Methods*, 47(1), 80-91. doi:10.1080/03610926.2017.1300282

Examples

```
Vmf(N1 = 37, N2 = 45, N = 16, m1 = 12, m2 = 8)
# Matches the exact variance from DISTM() when w = 1:
DISTM(N1 = 37, N2 = 45, N = 16, m1 = 12, m2 = 8, w = 1)$Vm
```

Index

BiasedUrn::dMWNCHypergeo(), [9](#)

BiasedUrn::dWNBHypergeo(), [9](#)

DISTM, [2](#)

DISTM(), [5](#), [10](#), [14](#)

estprop, [3](#)

estprop(), [8](#), [10](#)

inferW, [4](#)

inferW(), [6](#), [10](#)

likW, [6](#)

likW(), [5](#)

MHALG, [7](#)

MHALG(), [3](#), [10–12](#)

PROBM, [9](#)

PROBM(), [3](#), [4](#)

restable, [10](#)

restable(), [3](#), [8](#)

RMMHALG, [11](#)

RMMHALG(), [3](#), [10](#), [12](#), [13](#)

SMRMMHALG, [12](#)

SMRMMHALG(), [3](#), [10](#)

Vmf, [14](#)

Vmf(), [3](#)