Package: MCMCprecision (via r-universe)

October 29, 2026

Type Package

Title Precision of Discrete Parameters in Transdimensional MCMC

Version 0.4.2

Date 2025-07-22

Author Daniel W. Heck [aut, cre] (ORCID:
<https://orcid.org/0000-0002-6302-9252>)

Maintainer Daniel W. Heck <daniel.heck@uni-marburg.de>

Description Estimates the precision of transdimensional Markov chain Monte Carlo (MCMC) output, which is often used for Bayesian analysis of models with different dimensionality (e.g., model selection). Transdimensional MCMC (e.g., reversible jump MCMC) relies on sampling a discrete model-indicator variable to estimate the posterior model probabilities. If only few switches occur between the models, precision may be low and assessment based on the assumption of independent samples misleading. Based on the observed transition matrix of the indicator variable, the method of Heck, Overstall, Gronau, & Wagenmakers (2019, Statistics & Computing, 29, 631-643) <doi:10.1007/s11222-018-9828-0> draws posterior samples of the stationary distribution to (a) assess the uncertainty in the estimated posterior model probabilities and (b) estimate the effective sample size of the MCMC output.

License GPL-3

Encoding UTF-8

Depends R (>= 4.0.0)

Imports Rcpp, parallel, utils, stats, Matrix, combinat

Suggests testthat, R.rsp

LinkingTo Rcpp, RcppArmadillo, RcppProgress, RcppEigen

RoxygenNote 7.3.2

URL https://github.com/danheck/MCMCprecision

VignetteBuilder R.rsp

NeedsCompilation yes

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

Date/Publication 2025-07-22 09:20:08 UTC

RemoteUrl https://github.com/cran/MCMCprecision

RemoteRef HEAD

RemoteSha 23e017c53a8d7f2138697e05c9c2a8dfa79de403

Contents

MCMCprecision-package 2
best_models 3
fit_dirichlet 4
rdirichlet 5
rmarkov 5
stationary 6
stationary_mle 8
summary.stationary 9
transitions 10

Index **12**

`MCMCprecision-package` *MCMCprecision: Precision of discrete parameters in transdimensional MCMC*

Description

MCMCprecision estimates the precision of the posterior model probabilities in transdimensional Markov chain Monte Carlo methods (e.g., reversible jump MCMC or product-space MCMC). This is useful for applications of transdimensional MCMC such as model selection, mixtures with varying numbers of components, change-point detection, capture-recapture models, phylogenetic trees, variable selection, and for discrete parameters in MCMC output in general.

Details

The main function to assess the estimation uncertainty of discrete MCMC output is is `stationary`.

The method is explained in detail in Heck et al. (2019, Statistics & Computing), available in the package by calling: `vignette("MCMCprecision")`

Author(s)

Daniel W. Heck

References

Heck, D. W., Overstall, A. M., Gronau, Q. F., & Wagenmakers, E.-J. (2019). Quantifying uncertainty in transdimensional Markov chain Monte Carlo using discrete Markov models. Statistics & Computing, 29, 631–643. doi:10.1007/s1122201898280

See Also

Useful links:

- https://github.com/danheck/MCMCprecision

`best_models` *Precision for the k Best-Performing Models*

Description

Assesses the precision in estimating the ranking of the k best-performing models.

Usage

```
best_models(samples, k, ties.method = "min")
```

Arguments

`samples`	a matrix with posterior samples (one per row) for the model posterior probabilities (one model per column). Can be estimated using `stationary` with the argument `summary = FALSE`.
`k`	number of best-performing models to be considered
`ties.method`	a character string specifying how ties are treated, see `rank`

Examples

```
# a sequence of uncorrelated model indices
mult <- rmultinom(1000, 1, c(.05, .6, .15, .12, .08))
idx <- apply(mult, 2, which.max)
z <- letters[idx]
stat <- stationary(z, summary = FALSE)
best_models(stat, 3)
```

`fit_dirichlet` *Estimate Parameters of Dirichlet Distribution*

Description

C++ implementation of the fixed-point iteration algorithm by Minka (2000).

Usage

```
fit_dirichlet(x, const, maxit = 1e+05, abstol = 0.1)
```

Arguments

`x`	a matrix of Dirichlet samples, one row per observation.
`const`	constant that is added to avoid problems with zeros in `log(x)`. The default is `const = min(x[x>0])*.01`.
`maxit`	maximum number of iterations.
`abstol`	The absolute convergence tolerance: maximum of absolute differences of Dirichlet parameters.

Details

The algorithm is used to estimate the effective sample size based on samples of posterior model probabilities (see `stationary` and `summary.stationary`).

References

Minka, T. (2000). Estimating a Dirichlet distribution. Technical Report.

See Also

`rdirichlet`

Examples

```
x <- rdirichlet(100, c(8,1,3,9))
fit_dirichlet(x)
```

rdirichlet *Random Sample from Dirichlet Distribution*

Description

Random generation from the Dirichlet distribution.

Usage

```
rdirichlet(n, a)
```

Arguments

n	number of samples
a	vector or matrix of shape parameters

See Also

fit_dirichlet

Examples

```
rdirichlet(2, c(1,5,3,8))
```

rmarkov *Generate a sample of a discrete-state Markov chain*

Description

Generates a sequence of discrete states from a discrete-time Markov chain with transition matrix P.

Usage

```
rmarkov(n, P, start = rep(1, ncol(P)))
```

Arguments

n	length of the generated sequence.
P	transition matrix (rows are normalized to sum to 1).
start	vector with nonnegative values that defines the discrete starting distribution at t=0 (start is normalized to sum to 1). The default is a discrete uniform distribution.

Examples

```
P <- matrix(c(.30, .50, .20,
              .05, .25, .70,
              .00, .10, .90), 3, 3, byrow=TRUE)
rmarkov(50, P)
```

`stationary` *Precision of stationary distribution for discrete MCMC variables*

Description

Transdimensional MCMC methods include a discrete model-indicator variable z with a fixed but unknown stationary distribution with probabilities π (i.e., the model posterior probabiltiies). The function `stationary` draws posterior samples to assess the estimation uncertainty of π.

Usage

```
stationary(
  z,
  N,
  labels,
  sample = 1000,
  epsilon = "1/M",
  cpu = 1,
  method = "arma",
  digits = 6,
  progress = TRUE,
  summary = TRUE
)
```

Arguments

`z`	MCMC output for the discrete indicator variable with numerical, character, or factor labels (can also be a `mcmc.list` or a matrix with one MCMC chain per column).
`N`	the observed `transitions` matrix (if supplied, `z` is ignored). A quadratic matrix with sampled transition frequencies (`N[i,j]` = number of switches from `z[t]=i` to `z[t+1]=j`).
`labels`	optional: vector of labels for complete set of models (e.g., models not sampled in the chain `z`). If `epsilon=0`, this does not affect inferences due to the improper Dirichlet(0,..,0) prior.
`sample`	number of posterior samples to be drawn for the stationary distribution π.
`epsilon`	prior parameter for the rows of the estimated transition matrix P: $P[i,]$ ~ Dirichlet$(\epsilon, ..., \epsilon)$. The default `epsilon="1/M"` (with M = number of sampled models) provides estimates close to the i.i.d. estimates and is numerically stable. The alternative

epsilon=0 minimizes the impact of the prior and renders non-sampled models irrelevant. If method="iid" (ignores dependencies), a Dirichlet prior is assumed on the stationary distribution π instead of the rows of the transition matrix P.

cpu — number of CPUs used for parallel sampling. Will only speed up computations for large numbers of models (i.e., for large transition matrices).

method — how to compute eigenvectors:

- "arma" (default): Uses RcppArmadillo::eig_gen.
- "base": Uses base::eigen, which might be more stable, but also much slower than "arma" for small transition matrices.
- "eigen": Uses package RcppEigen::EigenSolver
- "armas": Uses sparse matrices with RcppArmadillo::eigs_gen, which can be faster for very large number of models if epsilon=0 (might be numerically unstable).
- "iid": Assumes i.i.d. sampling of the model indicator variable z. This is only implemented as a benchmark, because results cannot be trusted if the samples z are correlated (which is usually the case for transdimensional MCMC output)

digits — number of digits that are used for checking whether the first eigenvalue is equal to 1 (any difference must be due to low numerical precision)

progress — whether to show a progress bar (not functional for cpu>1)

summary — whether the output should be summarized. If FALSE, posterior samples of the stationary probabilities are returned.

Details

The method draws independent posterior samples of the transition matrix P for the discrete-valued indicator variable z (usually, a sequence of sampled models). For each row of the transition matrix, a Dirichlet$(\epsilon, ..., \epsilon)$ prior is assumed, resulting in a conjugate Dirichlet posterior. For each sample, the eigenvector with eigenvalue 1 is computed and normalized. These (independent) posterior samples can be used to assess the estimation uncertainty in the stationary distribution pi of interest (e.g., the model posterior probabilities) and to estimate the effective sample size (see summary.stationary).

Value

default: a summary for the posterior distribution of the model posterior probabilities (i.e., the fixed but unknown stationary distribution of z). If summary=FALSE, posterior samples for pi are returned.

See Also

best_models, summary.stationary

Examples

```
# data-generating transition matrix
P <- matrix(c(.1,.5,.4,
              0, .5,.5,
```

```
              .9,.1,0), ncol = 3, byrow=TRUE)

# input: sequence of sampled models
z <- rmarkov(500, P)
stationary(z)

# input: transition frequencies
N <- transitions(z)
samples <- stationary(N = N, summary = FALSE)

# summaries:
best_models(samples, k = 3)
summary(samples)
```

`stationary_mle` *MLE for stationary distribution of discrete MCMC variables*

Description

Maximum-likelihood estimation of stationary distribution π based on (a) a sampled trajectory z of a model-indicator variable or (b) a sampled transition count matrix N.

Usage

```
stationary_mle(z, N, labels, method = "rev", abstol = 1e-05, maxit = 1e+05)
```

Arguments

`z`	MCMC output for the discrete indicator variable with numerical, character, or factor labels (can also be a `mcmc.list` or a matrix with one MCMC chain per column).
`N`	the observed `transitions` matrix (if supplied, `z` is ignored). A quadratic matrix with sampled transition frequencies (`N[i,j]` = number of switches from `z[t]=i` to `z[t+1]=j`).
`labels`	optional: vector of labels for complete set of models (e.g., models not sampled in the chain `z`). If `epsilon=0`, this does not affect inferences due to the improper Dirichlet(0,..,0) prior.
`method`	Different types of MLEs: • `"iid"`: Assumes i.i.d. sampling of the model indicator variable `z` and estimates π as the relative frequencies each model was sampled. • `"rev"`: Estimate stationary distribution under the constraint that the transition matrix is reversible (i.e., fulfills detailed balance) based on the iterative fixed-point algorithm proposed by Trendelkamp-Schroer et al. (2015) • `"eigen"`: Computes the first left-eigenvector (normalized to sum to 1) of the sampled transition matrix
`abstol`	absolute convergence tolerance (only for `method = "rev"`)
`maxit`	maximum number of iterations (only for `method = "rev"`)

Details

The estimates are implemented mainly for comparison with the Bayesian sampling approach implemented in `stationary`, which quantify estimation uncertainty (i.e., posterior SD) of the posterior model probability estimates.

Value

a vector with posterior model probability estimates

References

Trendelkamp-Schroer, B., Wu, H., Paul, F., & Noé, F. (2015). Estimation and uncertainty of reversible Markov models. The Journal of Chemical Physics, 143(17), 174101. doi:10.1063/1.4934536

See Also

`stationary`

Examples

```
P <- matrix(c(.1,.5,.4,
              0,.5,.5,
              .9,.1,0), ncol = 3, byrow=TRUE)
z <- rmarkov(1000, P)
stationary_mle(z)

# input: transition frequency
tab <- transitions(z)
stationary_mle(N = tab)
```

---

summary.stationary *Summary for Posterior Model Probabilities*

Description

Summary for a sample of posterior model probabilities (`stationary`). Also provides the estimated effective sample size and summaries for all pairwise Bayes factors.

Usage

```
## S3 method for class 'stationary'
summary(object, BF = FALSE, logBF = FALSE, ...)
```

Arguments

object	posterior samples of the stationary distribution (rows = samples; columns = models).
BF	whether to compute summaries for all pairwise Bayes factors.
logBF	whether to summarize log Bayes factors instead of Bayes factors.
...	passed to fit_dirichlet to estimate effective sample size (e.g., maxit and abstol).

Details

Effective sample is estimated by fitting a Dirichlet model to the posterior model probabilities (thereby assuming that samples were drawn from an equivalent multinomial distribution based on independent sampling). More precisely, sample size is estimated by the sum of the Dirichlet parameters $\sum \alpha[i]$ minus the prior sample size $\epsilon * M^2$ (where M is the number of sampled models and ϵ the prior parameter for each transition frequency).

Value

a list with estimates for "pp" = posterior model probabilities, "n.eff" = effective sample size, and "bf" = pairwise Bayes factors (optional)

See Also

stationary, fit_dirichlet

Examples

```
P <- matrix(c(.1,.5,.4,
              0, .5,.5,
              .9,.1,0), ncol = 3, byrow=TRUE)
z <- rmarkov(1000, P)
samples <- stationary(z, summary = FALSE)
summary(samples)
```

---

transitions *Get matrix of observed transition frequencies*

Description

Summarizes a sequence of discrete values by the observed transition frequencies.

Usage

```
transitions(z, labels, order = 1)
```

Arguments

z	vector of model indices (numerical or character)
labels	fixed labels for models that should be included in transition matrix, e.g., labels=1:20 or c("m1","m2",...)
order	order of the transition table. If order=1, a matrix with transition frequencies from z[t+1] is returned. If order=2, a 3-dimensional array is returned with transition frequencies for z[t], z[t+1], and z[t+2].

Value

a square matrix with transition frequencies

Examples

```
P <- matrix(c(.9,.1,0,
              .1,.6,.3,
              .2,.3,.5), 3, byrow=TRUE)
z <- rmarkov(2000, P)
transitions(z)
transitions(z, order = 2)
```

Index

best_models, 3, *7*

eigen, *7*

fit_dirichlet, 4, *5*, *10*

mcmc.list, *6*, *8*
MCMCprecision *(*MCMCprecision-package*)*, 2
MCMCprecision-package, 2

rank, *3*
rdirichlet, *4*, 5
rmarkov, 5

stationary, *2–4*, 6, *9, 10*
stationary_mle, 8
summary.stationary, *4*, *7*, 9

transitions, *6*, *8*, 10