Package: Nmix (via r-universe)

October 26, 2026

Type Package

Title Bayesian Inference on Univariate Normal Mixtures

Version 2.0.5

Date 2023-09-19

Description A program for Bayesian analysis of univariate normal mixtures with an unknown number of components, following the approach of Richardson and Green (1997) <doi:10.1111/1467-9868.00095>. This makes use of reversible jump Markov chain Monte Carlo methods that are capable of jumping between the parameter sub-spaces corresponding to different numbers of components in the mixture. A sample from the full joint distribution of all unknown variables is thereby generated, and this can be used as a basis for a thorough presentation of many aspects of the posterior distribution.

Language en-GB

License GPL (>= 2)

NeedsCompilation yes

Author Peter Green [aut, cre] (<https://orcid.org/0000-0002-4367-4756>)

Maintainer Peter Green <P.J.Green@bristol.ac.uk>

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

Date/Publication 2023-09-19 22:30:33 UTC

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

RemoteRef HEAD

RemoteSha 22a27ea53cc5a71d185a6d7804630f529c9d51ef

Contents

Nmix-package 2
enz 3
galx 3

lnacid 4
Nmix 5
plot.nmix 8
print.nmix 9
readf2cio 10
sdrni 11
summary.nmix 12

Index **14**

`Nmix-package` *Bayesian Inference on Univariate Normal Mixtures*

Description

A program for Bayesian analysis of univariate normal mixtures, implementing the approach of Richardson and Green (1997) <doi:10.1111/1467-9868.00095>

Details

A program for Bayesian analysis of univariate normal mixtures with an unknown number of components, implementing the approach of Richardson and Green, Journal of the Royal Statistical Society, B, 59, 731-792 (1997); see also the correction in J. R. Statist. Soc. B, 1998, 60, 661). Computation is by reversible jump Markov chain Monte Carlo; this package is essentially an R interface to the Fortran program originally written in 1996 for the MCMC sampling, together with some facilities for displaying and summarising the resulting posterior distribution, and reporting the sampler performance.

Author(s)

Maintainer: NA

References

Richardson and Green (1997) <doi:10.1111/1467-9868.00095> (J. R. Statist. Soc. B, 59, 731-792; see also the correction in <doi:10.1111/1467-9868.00146>, J. R. Statist. Soc. B, 1998, 60, 661).

The author is grateful to Peter Soerensen for providing the interface to the C i/o routines used here.

Examples

```
data(galx)
z<-Nmix('galx',nsweep=10000,nburnin=1000)
print(z)
summary(z)
```

enz *Enzyme data set*

Description

Enzymatic activity in the blood, for an enzyme involved in the metabolism of carcinogenic substances, among a group of 245 unrelated individuals.

Usage

```
data("enz")
```

Format

The format is: num [1:245] 0.13 0.08 1.261 0.224 0.132 ...

Source

Bechtel, Y. C., Bonaiti-Pellik, C., Poisson, N., Magnette, J. and Bechtel, P. R. (1993) A population and family study of N-acetyltransferase using caffeine urinary metabolites. Clin. Pharm. Therp., 54, 134- 141.

References

Richardson and Green (J. R. Statist. Soc. B, 1997, 59, 731-792.

Examples

```
data(enz)
z<-Nmix('enz',nsweep=5000,nburnin=500,out="d")
```

galx *Galaxy data set*

Description

Velocities of 82 distant galaxies, diverging from our own galaxy

Usage

```
data("galx")
```

Format

The format is: num [1:82] 9.17 9.35 9.48 9.56 9.78 ...

Source

Roeder, K. (1990) Density estimation with confidence sets exemplified by superclusters and voids in the galaxies. J. Am. Statist. Ass., 85, 617-624.

References

Richardson and Green (J. R. Statist. Soc. B, 1997, 59, 731-792.

Examples

```
data(galx)
z<-Nmix('galx',nsweep=10000,nburnin=1000,out="d")
```

---

lnacid *Lake acidity data set*

Description

Acidity index measured in a sample of 155 lakes in north-central Wisconsin, on log scale.

Usage

```
data("lnacid")
```

Format

The format is: num [1:155] 2.93 3.91 3.73 3.69 3.82 ...

Source

Crawford, S. L., DeGroot, M. H., Kadane, J. B. and Small, M. J. (1992) Modeling lake chemistry distributions: approximate Bayesian methods for estimating a finite mixture model. Technometrics, 34, 441-453.

References

Richardson and Green (J. R. Statist. Soc. B, 1997, 59, 731-792.

Examples

```
data(lnacid)
z<-Nmix('lnacid',nsweep=10000,nburnin=1000,out="d")
```

Nmix *Bayesian Inference on Univariate Normal Mixtures*

Description

Wrapper for Nmix Fortran program that uses a Reversible jump Markov chain sampler to simulate from the posterior distribution of a univariate normal mixture model

Usage

```
Nmix(y,tag="",seed=0,nsweep=10000,nburnin=0,
 kinit=1,qempty=1,qprior=0,qunif=0,qfix=0,qrkpos=0,qrange=1,qkappa=0,qbeta=1,
 alpha=2,beta=0.02,delta=1,eee=0,fff=0,ggg=0.2,
 hhh=10,unhw=1.0,kappa=1.0,lambda=-1,xi=0.0,sp=1,
 out="Dkdep",nspace=nsweep%/%1000,
 nmax=length(y),ncmax=30,ncmax2=10,ncd=7,ngrid=200,k1k2=c(2,8),
 idebug=-1,qdebug=0)
```

Arguments

y	either (i) a numerical data vector, (ii) a character scalar naming a numerical data vector in the global environment or (iii) a character scalar identifying a file y.dat in the current working directory containing a dataset
tag	name for the dataset, in the case that y is a numerical vector
seed	positive integer to set random number seed for a reproducible run, or 0 to initialise this process; output value can be used to replicate run subsequently
nsweep	number of sweeps
nburnin	length of burn in
kinit	integer, initial number of components
qempty	integer, 1 or 0 according to whether the empty-component birth/death moves should be used
qprior	integer, 1 or 0 according to whether the prior should be simulated instead of the posterior
qunif	integer, 1 or 0 according to whether the uniform proposals should be used for the component means instead of gaussian ones
qfix	integer, 1 or 0 according to whether the number of components should be held fixed (at the value of kinit)
qrkpos	integer, 1 or 0 according to whether the the number of non-empty components should be reported throughout
qrange	integer, 1 or 0 according to whether range-based parameter priors should be used
qkappa	integer, 1 or 0 according to whether kappa should be updated
qbeta	integer, 1 or 0 according to whether beta should be updated
alpha	numeric, set value of parameter alpha

beta	numeric, set value of parameter beta
delta	numeric, set value of parameter delta
eee	numeric, set value of parameter e
fff	numeric, set value of parameter f
ggg	numeric, set value of parameter g
hhh	numeric, set value of parameter h
unhw	numeric, set value of half-width for uniform proposals
kappa	numeric, set value of parameter kappa
lambda	numeric, set value of parameter lambda; the value -1 (the default) means a prior for k uniform on 1,2,...ncmax
xi	numeric, set value of parameter xi
sp	numeric, set value of parameter s
out	character string to specify optional output: string containing letters 'D','C','A','p','k','d','e','a' (any others are ignored); "*" is equivalent to "DCApkeda". See Details.
nspace	spacing between samples recorded in time-series traces (see Details)
nmax	integer, set upper bound for n
ncmax	integer, set upper bound for k; the same as kmax in the references
ncmax2	integer, set upper bound for k in output components pe and avn
ncd	integer, set number of conditional densities computed
ngrid	integer, set number of grid points for density evaluation
k1k2	vector of 2 integers, set minimum and maximum number of components for classification calculation
idebug	integer, number of sweep from which to print debugging information
qdebug	integer 1 or 0 according to whether debugging information is to be printed

Details

Output options: Summaries

letter		output component
D	density	den
C	classification	pcl and scl
A	average component occupancy	avn

Traces

letter		component of traces
p	parameters	pars
k	number of components	k
d	deviance	deviance
e	entropy	entropy

a	allocations	`alloc`

Value

An object of class `nmix`. List with numerous components, including

`post`	posterior distribution of number of components `k`
`pe`	list whose `k`'th component is a `k` by 3 matrix of estimated posterior means of weights, means and sd's for a mixture with `k` components
`den`	matrix of density estimates for `k=1,2,...,6` and overall, preceded by row of abcissae at which they are evaluated - only when `out` includes "D"
`avn`	order-`ncmax2` square matrix with `(i,j)` entry the posterior expected number of observations allocated to component `i` when there are `j` components in the mixture - only when `out` includes "A"
`traces`	list of named vectors, traces of selected statistics `k`, `entropy` (as defined in Green and Richardson, 2001), etc, sub-sampled to every `nspace` sweeps
`iflag`	integer flagging successful completion of simulation (0) or not (1)

Author(s)

Peter J. Green

References

Richardson, S. and Green, P. J. On Bayesian analysis of mixtures with an unknown number of components (with discussion), J. R. Statist. Soc. B, 1997, 59, 731-792; see also the correction in J. R. Statist. Soc. B, 1998, 60, 661.

Green, P. J. and Richardson, S. Modelling heterogeneity with and without the Dirichlet process, Scandinavian Journal of Statistics, 2001, 28, 355-375.

The author is grateful to Peter Soerensen for providing the interface to the C i/o routines used here, borrowed from his package `qgg`.

Examples

```
data(galx)
z<-Nmix('galx',nsweep=10000,nburnin=1000,out="Dkd")
print(z)
summary(z)
plot(z)
```

plot.nmix *Plotting for Bayesian Inference on Univariate Normal Mixtures*

Description

Plotting of various information from nmix object on current graphics device

Usage

```
## S3 method for class 'nmix'
plot(x, which=1:5, separate=FALSE, plugin=FALSE, offset=1, nsamp=50,
 equi=TRUE, allsort=TRUE, trued=NULL, ...)
```

Arguments

x	nmix object, as output by Nmix function
which	integer vector, specifying which of several available plots are required, see 'Details' below
separate	logical, if TRUE opens a fresh default device for each new plot, otherwise prompts before overwriting a previous plot
plugin	logical, should plug-in estimator of density, computed from posterior means of parameters, be superimposed on density plot in darkgreen, in the case which contains 1?
offset	t numeric, vertical displacement between plotted traces, in the case which contains 2.
nsamp	integer, number of posterior samples plotted, in the case which contains 3.
equi	logical, should thinning of posterior density samples be equi-spaced, rather than random, in the case which contains 3
allsort	logical, should observations be sorted before making posterior clusters plot, in the case which contains 4
trued	vectorised function defining a probability density function to be superimposed in blue on density plots, in the cases which contains 1 and/or 3
...	additional arguments to plot

Details

If which includes 1, a 2-panel plot of which: the first is a histogram of the data, and if z has a component den (Nmix output option D), superimposed plots of the posterior density estimates, conditional on k=1,2,...,6 and unconditionally (in red); and the second a barplot of the estimated posterior distribution of k.

If which includes 2, a multiple trace plot of various statistics for a thinned subsample of the MCMC run, after burn-in. The statistics are the (named) components of the list z$traces that are numerical vectors, some or all of (i) the number of components k (Nmix output option k), (ii) the entropy (Nmix output option e), and (iii) the deviance (Nmix output option d), of the current sample. The traces

may be of different lengths, the horizontal scales in the plot are adjusted to span the length of the (post burn-in) MCMC run, regardless of these differences.

If `which` includes 3 (and `Nmix` output option p is present), a thinned sample of size `nsamp` from the posterior distribution of the density function, computed from a thinned sample of (weight, mean, sd) values generated in the posterior simulation.

If `which` includes 4 (and `Nmix` output option a is present), an image plot showing the posterior probabilities that the corresponding observations are in the same mixture component. Not recommended for large `n`.

If `which` includes 5 (and `Nmix` output option C is present), a 4-panel plot displaying Bayersian classifications based on the fitted model, analogous to Fig. 10 in the Richardson and Green paper. The 4 panels corresponding to conditioning on the 4 values of `k` most probable according to the posterior (among those for which the necessary posterior sample information has been recorded (see argument `k1k2` of `Nmix`), and excepting `k=1`).

Value

`NULL, invisibly`; plot method for class `nmix`. Function called for its side effect of plotting selected information about the fitted posterior distribution and sampler performance from `x` on the current graphics device

Author(s)

Peter J. Green

References

Richardson and Green (J. R. Statist. Soc. B, 1997, 59, 731-792; see also the correction in J. R. Statist. Soc. B, 1998, 60, 661)

Examples

```
data(galx)
z<-Nmix('galx',nsweep=10000,nburnin=1000,out="d")
plot(z,1:2)
```

`print.nmix` *Printing for Bayesian Inference on Univariate Normal Mixtures*

Description

Printing of various information about `nmix` object on current output

Usage

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

Arguments

x	nmix object, as output by Nmix function
...	additional arguments to print

Details

Currently the information printed consists of the estimated posterior for k and basic parameters of the MCMC simulation: number of sweeps, length of burnin, random number seed to replicate the run, and confirmation of which MCMC moves used (latters s,w,p,a,h,b standing for split/merge, weights, parameters, allocations, hyperparameters and birth/death).

Value

x, invisibly; print method for class nmix. Function called for its side effect of printing selected information from x

Author(s)

Peter J. Green

References

Richardson and Green (J. R. Statist. Soc. B, 1997, 59, 731-792; see also the correction in J. R. Statist. Soc. B, 1998, 60, 661)

Examples

```
data(galx)
z<-Nmix('galx',nsweep=10000,nburnin=1000,out="d")
z
```

---

readf2cio *Reading binary file of structured binary numerical data*

Description

Reading binary file of structured binary numerical data, for use in reading into R numerical data written from Fortran

Usage

```
readf2cio(fn,imax=Inf,verbose=FALSE)
```

Arguments

fn	character variable, path to file to be read.
imax	maximum number of list components to be read.
verbose	boolean, should the reading be reported?

Details

The function is designed to expedite the transfer of possibly large quantities of numeric information, in binary form, written, typically incrementally, in a Fortran routine called from R, without using the arguments to the function.

Assumed format for binary files holding lists, matrices or vectors of numeric data:
writable from Fortran via f2cio interface, readable in R using readBin
file structure supported: binary file, with integer(4), real(4) or double(8) data
first record: list: 0 0
matrix or vector: nc mode (mode = 1, 2 or 3 for integer(4), real(4) or double(8))
succeeding records, one per component of list or row of matrix:
list: number of items, mode as integers, followed by data for this component (note that modes can differ between but not within components)
matrix or vector: data for this row
one-column matrices are delivered as vectors

Value

numeric list, vector or matrix according to layout of information in `fn`; see Details.

Author(s)

Peter J. Green

Examples

```
data(galx)
z<-Nmix('galx',nsweep=10000,nburnin=1000,out="d")
z
```

`sdrni` *Random number initialiser, allowing retrospective replication*

Description

Front-end to standard R random number seed setter, allowing retrospective replication

Usage

```
sdrni(seed)
```

Arguments

`seed`	non-negative integer random number seed, often 0 for absolute re-initialisation as with `set.seed(NULL)`

Details

Using `sdrni` to initialise random number stream allows a decision to repeat a simulation exactly, presumably with additional outputs, need only be made after seeing results; see Examples

Value

`seed` if input value is positive, otherwise the value that if used in a subsequent call will deliver exactly the same random numbers

Author(s)

Peter J. Green

Examples

```
sdrni(0)
runif(5)
keep<-sdrni(0)
runif(5)
sdrni(keep)
runif(5)
```

`summary.nmix` *Summary for Bayesian Inference on Univariate Normal Mixtures*

Description

Printing of various information from `nmix` object on current output

Usage

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

Arguments

`object`	`nmix` object, as output by `Nmix` function
`...`	additional arguments to `summary`

Details

Currently the information printed consists of the estimated posterior for `k` and basic parameters of the MCMC simulation: number of sweeps, length of burnin, random number seed to replicate the run, and confirmation of which MCMC moves used, and acceptance statistics for each type of trans-dimensional move.

Value

`object`, invisibly; summary method for class `nmix`. Function called for its side effect of printing selected information from `object`

Author(s)

Peter J. Green

References

Richardson and Green (J. R. Statist. Soc. B, 1997, 59, 731-792; see also the correction in J. R. Statist. Soc. B, 1998, 60, 661)

Examples

```
data(galx)
z<-Nmix('galx',nsweep=10000,nburnin=1000,out="d")
summary(z)
```

Index

∗ **datasets**
enz, 3
galx, 3
lnacid, 4
∗ **package**
Nmix-package, 2

enz, 3

galx, 3

lnacid, 4

Nmix, 5
Nmix-package, 2

plot, *8*
plot.nmix, 8
print, *10*
print.nmix, 9

readf2cio, 10

sdrni, 11
summary, *12*
summary.nmix, 12