

Package ‘Rtwalk’

January 20, 2025

Version 1.8.0

Date 2015-09-16

Title The R Implementation of the 't-walk' MCMC Algorithm

Author J Andres Christen

Maintainer J Andres Christen <jac@cimat.mx>

Depends R (>= 2.8.0)

Description The 't-walk' is a general-purpose MCMC sampler for arbitrary continuous distributions that requires no tuning.

License GPL-3

URL <http://www.cimat.mx/~jac/twalk/>

NeedsCompilation no

Repository CRAN

Date/Publication 2015-09-18 08:58:26

Contents

Ana	2
OneMove	3
PlotHist	5
PlotLogObj	6
Rtwalk	7
Runtwalk	8
SaveOutput	11
TS	12
Index	14

Ana *Perform some basic autocorrelation analysis.*

Description

Perform some basic autocorrelation analysis of the twalk MCMC output.

Usage

```
Ana(info, from=1, to=info$Tr, par=0, file="")
```

Arguments

info	as returned from Runtwalk.
from	iteration number to start plotting (from=0 begins at initialization point).
to	last iteration to plot.
par	parameter to analyze.
file	name of file to write results to (if not "").

Value

NULL

Author(s)

J Andres Christen (CIMAT, Guanajuato, MEXICO).

References

Christen JA and Fox C (2010). A general purpose sampling algorithm for continuous distributions (the t-walk)., Bayesian Analysis, 5 (2), 263-282. URL: <http://ba.stat.cmu.edu/journal/2010/vol05/issue02/christen.pdf>

See Also

[Runtwalk](#) for running the twalk.

Examples

```
#### We first load the twalk package:
library(Rtwalk)

#### A ver simple inline example, 4 independent normals N(0,1):
#####      dimension, num of it, -log of objective function besides a const, support,
info <- Runtwalk( dim=4, Tr=1000, Obj=function(x) { sum(x^2)/2 }, Supp=function(x) { TRUE },
x0=runif(4, min=20, max=21), xp0=runif(4, min=20, max=21))
```

```
#### and two (intentionally bad) initial points

### Perform some basic autocorrelation analysis
Ana( info, from=500)
```

OneMove

One move of the t-walk

Description

Evaluates the t-walk kernel once and returns the proposed jumping points and the acceptance probability.

Usage

```
OneMove(dim, Obj, Supp, x, U, xp, Up,
        at=6, aw=1.5, pphi=min(dim, 4)/dim, F1=0.4918, F2=0.9836, F3=0.9918
        , ...)
```

Arguments

dim	dimension of the objective function.
Obj	a function that takes a vector of length=dim and returns -log of the objective function, besides an adding constant.
Supp	a function that takes a vector of length=dim and returns TRUE if the vector is within the support of the objective and FALSE otherwise. Supp is <i>*always*</i> called right before Obj.
x	First of a pair of initial points, within the support of the objective function. x0 and xp must be different.
U	Current value of Obj at x.
xp	Second of a pair of initial points, within the support of the objective function. x0 and xp must be different.
Up	Current value of Obj at xp.
at	The remaining parameters are the traverse and walk kernel parameters, the parameter choosing probability and the cumulative probabilities of choosing each kernel. These are not intended to be modified in standard calculations.
aw	See description for at.
pphi	See description for at.
F1	See description for at.
F2	See description for at.
F3	See description for at.
...	Other parameters passed to Obj.

Value

A list with the following items:

y, yp proposals.

propU, propUp value of the objective at y and yp.

A Metropolis-Hastings ratio, acceptance probability = $\min(1, \text{move\$A})$.

funh Kernel used: 1=traverse, 2=walk, 3=hop, 4=blow.

Author(s)

J Andres Christen (CIMAT, Guanajuato, MEXICO).

References

Christen JA and Fox C (2010). A general purpose sampling algorithm for continuous distributions (the t-walk)., Bayesian Analysis, 5 (2), 263-282. URL: <http://ba.stat.cmu.edu/journal/2010/vol05/issue02/christen.pdf>

See Also

[Runtwalk](#)

Examples

```
#### We first load the twalk package:
library(Rtwalk)

#### A ver simple example, 4 independent normals N(0,1):
x <- runif( 4, min=20, max=21)
xp <- runif( 4, min=20, max=21)
U <- sum(x^2)/2
Up <- sum(xp^2)/2

move <- OneMove( dim=4, Obj=function(x) { sum(x^2)/2 }
  , Supp=function(x) { TRUE }, x=x, U=U, xp=xp, Up=Up)

if (runif(1) < move$A) #### the actual acceptance probability is min(1,A)
{ ## accepted
  x <- move$y
  U <- move$propU

  xp <- move$yp
  Up <- move$propUp
}
##else Not accepted

### etc.
```

PlotHist *Plot a histogram of a*

Description

Plot a histogram of the twalk MCMC output.

Usage

```
PlotHist( info, par=1, from=0, xlab=paste("Parameter", par), main="", ...)
```

Arguments

info	as returned from Runtwalk.
par	parameter number to plot.
from	iteration number to start plotting (from=0 begins at initialization point).
xlab	parameter passed to hist.
main	parameter passed to hist.
...	parameters passed to hist.

Value

hist object.

Author(s)

J Andres Christen (CIMAT, Guanajuato, MEXICO).

References

Christen JA and Fox C (2010). A general purpose sampling algorithm for continuous distributions (the t-walk)., Bayesian Analysis, 5 (2), 263-282. URL: <http://ba.stat.cmu.edu/journal/2010/vol05/issue02/christen.pdf>

See Also

[Runtwalk](#) For running the twalk.

Examples

```
#### We first load the twalk package:
library(Rtwalk)

#### A ver simple inline example, 4 independent normals N(0,1):
#####      dimension, num of it, -log of objective function besides a const, support,
```

```
info <- Runtwalk( dim=4, Tr=1000, Obj=function(x) { sum(x^2)/2 }, Supp=function(x) { TRUE },
x0=runif(4, min=20, max=21), xp0=runif(4, min=20, max=21))
#### and two (intentionally bad) initial points

### One can plot some histograms:
PlotHist(info, par=3)
```

PlotLogObj

Plot a trace of the Log of Objective function values

Description

Plot a trace of the Log of Objective function values, burn-in for convergence evaluation purposes.

Usage

```
PlotLogObj(info, from=0, to=length(info$Us))
```

Arguments

info	as returned from Runtwalk.
from	iteration number to start plotting (from=0 begins at initialization point).
to	last iteration to plot.

Value

NULL

Author(s)

J Andres Christen (CIMAT, Guanajuato, MEXICO).

References

Christen JA and Fox C (2010). A general purpose sampling algorithm for continuous distributions (the t-walk)., Bayesian Analysis, 5 (2), 263-282. URL: <http://ba.stat.cmu.edu/journal/2010/vol05/issue02/christen.pdf>

See Also

[Runtwalk](#) for running the twalk.

Examples

```
#### We first load the twalk package:
library(Rtwalk)

#### A ver simple inline example, 4 independent normals N(0,1):
#####      dimension, num of it, -log of objective function besides a const, support,
info <- Runtwalk( dim=4, Tr=1000, Obj=function(x) { sum(x^2)/2 }, Supp=function(x) { TRUE },
x0=runif(4, min=20, max=21), xp0=runif(4, min=20, max=21))
#### and two (intentionally bad) initial points

### or plot the log of the objective
PlotLogObj(info)
```

Rtwalk

The R implementation of the 't-walk'

Description

The 't-walk' is "A General Purpose Sampling Algorithm for Continuous Distributions", see <http://www.cimat.mx/~jac/twalk/> for technical details.

Details

Package: Rtwalk
Type: Package
Version: 1.0
Date: 2009-10-01
License: GPL (>= 2)
Copyright: (c) 2004-2009 J Andr es Christen (CIMAT, Guanajuato, MEXICO)
URL: <http://www.cimat.mx/~jac/twalk/>

The t-walk is run with the function:

Runtwalk

Please look at the documentation of the Runtwalk function for more details, or download the file <http://www.cimat.mx/~jac/twalk/examples.R>.

Author(s)

J Andres Christen (CIMAT, Guanajuato, MEXICO).

See Also

[Runtwalk](#)

Runtwalk

*Run the 't-walk'***Description**

Runs the 't-walk' and returns a list including the samples.

Usage

```
Runtwalk(Tr, dim = length(x0), Obj, Supp, x0, xp0
, PlotObj=FALSE, PlotLogPost=TRUE, dynty="b", pathcol="grey"
, add=FALSE, at=6, aw=1.5, pphi=min( dim, 4)/dim
, F1=0.4918, F2=F1+0.4918, F3=F2+0.0082, ...)
```

Arguments

Tr	number of iterations.
dim	dimension of the objective function.
Obj	a function that takes a vector of length dim and returns -log of the objective function, up to a constant.
Supp	a function that takes a vector of length=dim and returns TRUE if the vector is within the support of the objective and FALSE otherwise. Supp is <i>*always*</i> called right before Obj.
x0	First of a pair of initial points, within the support of the objective function. x0 and xp must be different.
xp0	Second of a pair of initial points, within the support of the objective function. x0 and xp must be different.
PlotObj	Some parameters for plotting the path of the twalk when dim=2. Only used for demonstration purposes, commonly PlotObj=FALSE and the rest is ignored. Set PlotLogPost=FALSE to also avoid plotting the LogPosterior time series as the talk progresses. This will force the twalk (if dim=2, add PlotObj=FALSE) to run with no graphics (eg. server or batch mode).
PlotLogPost	See description for PlotObj.
dynty	See description for PlotObj.
pathcol	See description for PlotObj.
add	See description for PlotObj.
at	The remaining parameters are the traverse and walk kernel parameters, the parameter choosing probability and the cumulative probabilities of choosing each kernel. These are not intended to be modified in standard calculations.
aw	See description for at.
pphi	See description for at.
F1	See description for at.
F2	See description for at.
F3	See description for at.
...	Other parameters passed to Obj.

Value

A list with the following items:

n	dimension of the objective.
Tr	number of iterations.
Us	value of -log of Obj for x at each iteration.
Ups	value of -log of Obj for xp at each iteration.
output	a TrXn matrix with the iterations for x.
outputp	a TrXn matrix with the iterations for xp.

Author(s)

J Andres Christen (CIMAT, Guanajuato, MEXICO).

References

Christen JA and Fox C (2010). A general purpose sampling algorithm for continuous distributions (the t-walk)., Bayesian Analysis, 5 (2), 263-282. URL: <http://ba.stat.cmu.edu/journal/2010/vol05/issue02/christen.pdf>

See Also

[OneMove](#) for performing a stand alone iteration of the t-walk kernel, to be inserted within a more complex MCMC with other transition kernels.

Examples

```
#### We first load the twalk package:
library(Rtwalk)

#### A ver simple inline example, 4 independent normals N(0,1):
#####      dimension, num of it, -log of objective function besides a const, support,
info <- Runtwalk( dim=4, Tr=1000, Obj=function(x) { sum(x^2)/2 }, Supp=function(x) { TRUE },
x0=runif(4, min=20, max=21), xp0=runif(4, min=20, max=21))
##### and two (intentionally bad) initial points

### One can plot some histograms:
PlotHist(info, par=3)
### Or time series of the parameters
TS(info)
### or plot the log of the objective
PlotLogObj(info)
### and remove the burn-in
PlotLogObj(info, from=500)
PlotHist(info, par=3, from=500)
TS( info, from=500)
### And do some basic autocorrelation analysis
Ana( info, from=500)
```

```

### And save the output as columns in a table
SaveOutput( info, file="Tsttwalk.dat")
### SaveOutput is simply a wrapper to the write.table function

##### A more complex Objective,
##### the posterior of alpha (shape) and beta (rate) in gamma sampling
##### The prior for alpha is U( 1, 4) and for beta is Exp(1)

### a initialization function
GaSamInit <- function(sample.size=100) {

### Set the dimension as the global variable npars
npars <- 2 ## alpha and beta

### sample 100 gammas with the true parameters 2.5 and 3
m <- sample.size ### sample size, now global variable m
smpl <- rgamma( sample.size, shape=2.5, rate=3)

### calculate the suff. statistics
r1 <- sum(smpl)
r2 <- sum(log(smpl))
}

### This is the -log of the posterior, -log of the objective
GaSamU <- function(x) {

al <- x[1]
be <- x[2]

### It is VERY advisable to try to do the calculations inside -log post:
-1*m*al*log(be) + m*lgamma(al) + (1-al)*r2 + be*(1+r1)
}

### This is the support:
GaSamSupp <- function(x) {

(((0 < x[1]) & (x[1] < 4)) & (0 < x[2]))
}

### Is also very advisable to have a function that generates initial (random?) points
### anything "within the same galaxy of the objective" most probably work
### for example, sample from the prior
GaSamX0 <- function(x) { c( runif(1, min=1, max=4), rexp(1,rate=1)) }

### The twalk is run with
### Don't forget to initialize the problem:
GaSamInit()
info <- Runtwalk( dim=npars, Tr=1000, Obj=GaSamU, Supp=GaSamSupp, x0=GaSamX0(), xp0=GaSamX0())

### This no longer works!!!

```

```
### Value of dim taken from the global var n
#n <- npars
#info <- Runtwalk( Tr=1000, Obj=GaSamU, Supp=GaSamSupp, x0=GaSamX0(), xp0=GaSamX0())

### See this and many more examples in: \url{http://www.cimat.mx/~jac/twalk/examples.R}
```

SaveOutput

Save the twalk MCMC output to a file.

Description

Save the twalk MCMC output to a file.

Usage

```
SaveOutput( info, file, pars=1:(info$dim), from=1, to=info$Tr,
row.names=FALSE, col.names=paste("X", pars), ...)
```

Arguments

info	as returned from Runtwalk.
file	name of file to write results to (of not "").
from	iteration number to start saving (from=0 begins at initialization point).
to	last iteration to save.
pars	parameters to save (defaults to all).
row.names, col.names, ...	parameters passed to write.table.

Value

write.table object.

Author(s)

J Andres Christen (CIMAT, Guanajuato, MEXICO).

References

Christen JA and Fox C (2010). A general purpose sampling algorithm for continuous distributions (the t-walk)., Bayesian Analysis, 5 (2), 263-282. URL: <http://ba.stat.cmu.edu/journal/2010/vol05/issue02/christen.pdf>

See Also

[Runtwalk](#) for running the twalk.

Examples

```
#### We first load the twalk package:
library(Rtwalk)

#### A ver simple inline example, 4 independent normals N(0,1):
#####      dimension, num of it, -log of objective function besides a const, support,
info <- Runtwalk( dim=4, Tr=1000, Obj=function(x) { sum(x^2)/2 }, Supp=function(x) { TRUE },
x0=runif(4, min=20, max=21), xp0=runif(4, min=20, max=21))
#### and two (intentionally bad) initial points

### Save the twalk MCMC output as columns in a table
SaveOutput( info, file="Tsttwalk.dat")
```

TS

Plot a 'time series' of the twalk MCMC output.

Description

Plot a 'time series' of the twalk MCMC output.

Usage

```
TS(info, pars=1:(info$dim), from=1, to=info$Tr, prime=FALSE)
```

Arguments

info	as returned from Runtwalk.
pars	parameter list to plot.
from	iteration number to start plotting (from=0 begins at initialization point).
to	last iteration to plot.
prime	plot xp (x') instead.

Value

plot object.

Author(s)

J Andres Christen (CIMAT, Guanajuato, MEXICO).

References

Christen JA and Fox C (2010). A general purpose sampling algorithm for continuous distributions (the t-walk)., Bayesian Analysis, 5 (2), 263-282. URL: <http://ba.stat.cmu.edu/journal/2010/vol05/issue02/christen.pdf>

See Also

[Runtwalk](#) For running the twalk.

Examples

```
#### We first load the twalk package:  
library(Rtwalk)
```

```
#### A ver simple inline example, 4 independent normals N(0,1):  
##### dimension, num of it, -log of objective function besides a const, support,  
info <- Runtwalk( dim=4, Tr=1000, Obj=function(x) { sum(x^2)/2 }, Supp=function(x) { TRUE },  
x0=runif(4, min=20, max=21), xp0=runif(4, min=20, max=21))  
##### and two (intentionally bad) initial points
```

```
### One can plot a time series of the parameters  
TS(info)
```

Index

- * **Autocorrelations**
 - Ana, [2](#)
 - * **Automatic**
 - OneMove, [3](#)
 - Runtwalk, [8](#)
 - * **IAT**
 - Ana, [2](#)
 - * **Log Objective**
 - PlotLogObj, [6](#)
 - * **MCMC**
 - Ana, [2](#)
 - OneMove, [3](#)
 - PlotHist, [5](#)
 - PlotLogObj, [6](#)
 - Runtwalk, [8](#)
 - SaveOutput, [11](#)
 - TS, [12](#)
 - * **Output**
 - SaveOutput, [11](#)
 - * **Plot**
 - PlotLogObj, [6](#)
 - * **Time series**
 - TS, [12](#)
 - * **output**
 - PlotHist, [5](#)
 - * **package**
 - Rtwalk, [7](#)
 - * **self adjusting**
 - OneMove, [3](#)
 - Runtwalk, [8](#)
 - * **stand alone kernel**
 - OneMove, [3](#)
 - * **twalk**
 - PlotHist, [5](#)
 - TS, [12](#)
- Ana, [2](#)
- OneMove, [3](#), [9](#)
- PlotHist, [5](#)
- PlotLogObj, [6](#)
- Rtwalk, [7](#)
- Runtwalk, [2](#), [4-7](#), [8](#), [11](#), [13](#)
- SaveOutput, [11](#)
- TS, [12](#)