Package ‘ars’

September 6, 2024

Title Adaptive Rejection Sampling

Version 0.8

Date 2024-09-05

Maintainer Paulino Perez Rodriguez <perpdgo@colpos.mx>

Depends R (>= 3.1.2)

Description Adaptive Rejection Sampling, Original version.

License GPL (>= 2)

NeedsCompilation yes

Author Paulino Perez Rodriguez [aut, cre] (Original C++ code from Arnost Komarek based on ars.f written by P. Wild and W. R. Gilks)

Repository CRAN

Date/Publication 2024-09-06 15:30:05 UTC

Contents

ars 1

Index **4**

`ars` *Adaptive Rejection Sampling*

Description

Adaptive Rejection Sampling from log-concave density functions

Usage

```
ars(n=1,f,fprima,x=c(-4,1,4),ns=100,m=3,emax=64,lb=FALSE,ub=FALSE,xlb=0,xub=0,...)
```

Arguments

`n`	sample size
`f`	function that computes log(f(u,...)), for given u, where f(u) is proportional to the density we want to sample from
`fprima`	d/du log(f(u,...))
`x`	some starting points in wich log(f(u,...) is defined
`ns`	maximum number of points defining the hulls
`m`	number of starting points
`emax`	large value for which it is possible to compute an exponential
`lb`	boolean indicating if there is a lower bound to the domain
`xlb`	value of the lower bound
`ub`	boolean indicating if there is a upper bound to the domain
`xub`	value of the upper bound bound
`...`	arguments to be passed to f and fprima

Details

ifault codes, subroutine initial

0: successful initialisation

1: not enough starting points

2: ns is less than m

3: no abscissae to left of mode (if lb = false)

4: no abscissae to right of mode (if ub = false)

5: non-log-concavity detect

ifault codes, subroutine sample

0: successful sampling

5: non-concavity detected

6: random number generator generated zero

7: numerical instability

Value

a sampled value from density

Author(s)

Paulino Perez Rodriguez, original C++ code from Arnost Komarek based on ars.f written by P. Wild and W. R. Gilks

References

Gilks, W.R., P. Wild. (1992) Adaptive Rejection Sampling for Gibbs Sampling, *Applied Statistics* **41**:337–348.

Examples

```
library(ars)

#Example 1: sample 20 values from the normal distribution N(2,3)
f<-function(x,mu=0,sigma=1){-1/(2*sigma^2)*(x-mu)^2}
fprima<-function(x,mu=0,sigma=1){-1/sigma^2*(x-mu)}
mysample<-ars(20,f,fprima,mu=2,sigma=3)
mysample
hist(mysample)

#Example 2: sample 20 values from a gamma(2,0.5)
f1<-function(x,shape,scale=1){(shape-1)*log(x)-x/scale}
f1prima<-function(x,shape,scale=1) {(shape-1)/x-1/scale}
mysample1<-ars(20,f1,f1prima,x=4.5,m=1,lb=TRUE,xlb=0,shape=2,scale=0.5)
mysample1
hist(mysample1)

#Example 3: sample 20 values from a beta(1.3,2.7) distribution
f2<-function(x,a,b){(a-1)*log(x)+(b-1)*log(1-x)}
f2prima<-function(x,a,b){(a-1)/x-(b-1)/(1-x)}
mysample2<-ars(20,f2,f2prima,x=c(0.3,0.6),m=2,lb=TRUE,xlb=0,ub=TRUE,xub=1,a=1.3,b=2.7)
mysample2
hist(mysample2)
```

Index

∗ **distribution**

ars, 1

ars, 1