Package 'voigt'

May 8, 2026

Type Package

Title The Voigt Distribution

Version 2.0

Date 2025-08-18

Maintainer Massimo Cannas <massimo.cannas@unica.it>

Description Random generation, density function and parameter estimation for the Voigt distribution. The main objective of this package is to provide R users with efficient estimation of Voigt parameters using classic iid data in a Bayesian framework. The estimating function allows flexible prior specification, specification of fixed parameters and several options for Markov Chain Monte Carlo posterior simulation. A basic version of the algorithm is described in: Cannas M. and Piras, N. (2025) <doi:10.1007/978-3-031-96303-2_53>.

Depends R (>= 3.5.0)

Imports invgamma, coda, pracma

License GPL-2

NeedsCompilation no

Author Massimo Cannas [aut, cre],
Nicola Piras [aut],
Daniele Chiriu [ctb]

Repository CRAN

Date/Publication 2025-08-18 16:00:03 UTC

Contents

voigt-package *Tools for working with the Voigt distribution*

Description

Random generation, density function and parameter estimation for the Voigt distribution. The main objective of this package is to provide R users with efficient estimation of Voigt parameters using classic iid data in a Bayesian framework. The estimating function allows flexible prior specification, specification of fixed parameters and several options for Markov Chain Monte Carlo posterior simulation. A basic version of the algorithm is described in: Cannas M. and Piras, N. (2025) <doi:10.1007/978-3-031-96303-2_53>.

Details

The package contains functions for working with the Voigt distribution in the R environment:

`rvoigt` generates random variates

`dvoigt` calculates voigt density

`evoigt` estimates parameters following a Bayesian approach

otherwise unavailable within R (R < 4.4.1). The main function `evoigt` provides both point and interval estimates for the Voigt parameters using a Bayesian approach.

Author(s)

Massimo Cannas [aut, cre], Nicola Piras [aut], Daniele Chiriu [ctb]

Maintainer: Massimo Cannas <massimo.cannas@unica.it>

References

Kendall, D. G. (1938). The effect of radiation damping and Doppler broadening on the atomic absorption coefficient. *Zeitschrift fur Astrophysik, Vol. 16, p.308* https://adsabs.harvard.edu/full/1938ZA.....16..308K

Cannas, M. and Piras, N. Mixture representation and parameter estimation for the Voigt profile (*submitted*)

Examples

```
## Not run:
##  See examples in the help file of each function.

## End(Not run)
```

evoigt *Estimation of Voigt distribution parameters*

Description

The function provides both point and interval estimates for the Voigt distribution parameters in a Bayesian way (see Details).

Usage

```
evoigt(data, hyper = NULL, init = NULL, S = 10000, burn = S/2,
thin = 10, fix.arg = NULL, chain = FALSE, ...)
```

Arguments

data	A numeric vector of length at least one containing only finite values.
hyper	A numeric vector of length six giving the hyperparameters for mu, sigma and gamma priors (in this order). If NULL all hyperparameters are set to 0.1.
init	The starting values for the chain parameters. If NULL (default) all chains start at 1. If *scalar* it is used as the common starting value. If *character* it can be set either to "rand" for random starting values or to "datadriven" for quantile based estimation of starting values.
S	The number of iterations in the Gibbs sampler. Default to 10000.
burn	The number of initial chain values to be discarded (*burn-in* period). Default to $S/2$.
thin	This parameter allows the user to specify if and how the chain should be thinned after burn-in. By default thin = 5 is used, which corresponds to keeping 1/5 of the chain values.
fix.arg	An optional vector of length 3 giving the values of fixed parameters of the Voigt distribution, in this order: μ, σ, γ with NA corresponding to an unknown parameter. By default all parameters are estimated. Parameters with fixed value are thus NOT estimated by the Gibbs sampler procedure.
chain	logical; if TRUE the output contains also the (thinned) chains values after burn-in. The first value of the chains is set according to the init argument.
...	Additional arguments to be passed to functions of the package coda. A prob argument can be used to specify the required probability of the credibility intervals via the HPDinterval function (default is 0.95; see the examples). Note that if S is too small the final probability can be different from the required one .

Details

The function runs a Gibbs sampler for estimating parameters using the following prior distributions: $\mu \sim N(\mu_0, \nu_0^2)$, $\sigma \sim \Gamma(a, b)$ and $\gamma \sim \Gamma(c, d)$.

Value

`evoigt` returns a list with the following components:

`posterior mean`	the vector of posterior means
`posterior median`	the vector of posterior medians
`HPD interval`	the highest probability density intervals for the parameters
`chain`	the thinned chain values after burn-in (only if argument `chain` is TRUE.)

Author(s)

Massimo Cannas [aut, cre], Nicola Piras [aut], Daniele Chiriu [ctb]

References

Cannas, M. and Piras, N. (2025) Estimation of Voigt Distribution Parameters: A Bayesian Approach. *(conference paper)* https://link.springer.com/chapter/10.1007/978-3-031-96303-2_53

Cannas, M. and Piras, N. Mixture representation and parameter estimation for the Voigt profile (*submitted*)

See Also

See `HPDinterval` for details on how the highest posterior density interval is built.

Examples

```
x = rvoigt(500, mu=0, sigma=1,gamma=1)
# point estimates and (default) 95% credibility intervals
evoigt(data=x)
# point estimates and 90% credibility intervals
evoigt(data=x, prob = 0.9)
# chain values
res = evoigt(data=x, prob = 0.9, chain=TRUE)
mu.chain = res$mu.chain
plot(0: (length(mu.chain)-1), type = "l", mu.chain, xlab="",ylab=c(expression(mu)) )
points(0,mu.chain[1],pch=1,col="red")
# if a parameter is known its value can be fixed using "fix.arg"
# e.g. set sigma =1:
res = evoigt(data=x, fix.arg=c(NA, 1, NA))
res["posterior mean"]
res["posterior median"]
res["HPD interval"]

# Inverse sampling from histogram
## Not run:
x = seq(-15000,15000, by=0.1)
y = dvoigt(x, sigma=2, gamma=3, fast=FALSE)
int<- 1:length(x)
breaks <-  x[- length(x)] + (x[-1] - x[-length(x)] ) /2
```

```
breaks = c( x[1]- (x[2]-x[1])/2 ,breaks, x[length(x)]+ (x[length(x)]-x[length(x)-1])/2)
bins <- sample(int, size=5000, prob=y, replace=TRUE)
samplehist <- vector()
for (j in 1:length(bins)){
samplehist[ j ] = runif(1, min= breaks[ bins[ j ] ], max= breaks[ bins[ j ] +1 ])
}
# estimation using sample histogram
evoigt(samplehist)
## End(Not run)
```

---

raman *Raman spectrum for ochre data*

Description

Excerpt of data set used by Pisu et al.(2025) *Innovative method for provenance studies in cultural heritage: A new algorithm based on observables from high-resolution Raman spectra of red ochre* to analyze the Raman spectrum of red ochre, typically derived from iron oxides.

Usage

```
data("raman")
```

Format

A data frame with 2048 observations on 2 variables.

- `x` Points along the x-axis for which the empirical density is available in y.
- `y` Empirical density.

Details

The data set is used in the example section to illustrate the use of function `evoigt`.

Source

Pisu et al. Innovative method for provenance studies in cultural heritage: A new algorithm based on observables from high-resolution Raman spectra of red ochre, *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy, 329 (2025)*

See Also

See also `evoigt`

Examples

```
data(raman)
plot(raman$x,raman$y,type="l", main="")

# select fourth peak
abline(v=raman$x[782],col="red")
abline(v=raman$x[1097],col="red")
datapeak = raman[(782):(1097),]
x <- datapeak[,1]
y <- datapeak[,2]
plot(x,y,type="l", main="")

# centering and scaling
offset<-451.1; A<- 27294.56; medianx <- 407.30
x= x - medianx
y = (y-offset)/A
smooth <- smooth.spline(x, y, spar = 0.5)
ys <- predict(smooth, x)$y

# inverse sampling from empirical profile (x,y)
int<- 1:length(x)
breaks <-  x[- length(x)] + (x[-1] - x[-length(x)] ) /2
breaks = c( x[1]- (x[2]-x[1])/2 ,breaks, x[length(x)]+ (x[length(x)]-x[length(x)-1])/2)
samplesize=5000
bins <- sample(int, size=samplesize, prob=ys, replace=TRUE)
samplehist <- vector()
for (j in 1:length(bins)){
  samplehist[ j ] = runif(1, min= breaks[ bins[ j ] ], max= breaks[ bins[ j ] +1 ])
}

# estimate using Gibbs
## Not run:
res <- evoigt(samplehist)
# classic parameters
# mu, sigma and gamma point estimates
classic.par<-unlist(res["posterior mean"])
classic.par
# physical parameters
# w_G = sqrt(8 log 2) *sigma; w_L = 2* gamma
phy.par<- classic.par*c(0, sqrt(8*log(2)),  2)
names(phy.par)<-c("mu","w_G", "w_L")
phy.par

# plot estimated voigt versus empirical profile (original)
plot(x + medianx, y*A+offset,type="l",lwd=2,col="black",xlab="",ylab="")
lines(x+medianx,dvoigt(x,sigma=classic.par[2],gamma=classic.par[3])*A+offset,
col="red",lwd=3,lty=1)
legend("topright",legend = c("empirical profile","fitted profile"),
col = c("black", "red"),lty = c(1, 1),lwd = c(2, 3),bty = "n")

# R^2
yhat <- dvoigt(x,sigma = classic.par[2],gamma = classic.par[3])*A+offset
```

```
my<-mean(ys*A+offset)
1- sum(ys*A+offset-yhat)^2/sum((ys*A+offset-my)^2)

## End(Not run)
```

Voigt *The Voigt Distribution*

Description

Density function and random generation for the Voigt distribution with median equal to mu and scale parameters equal to sigma and gamma.

Usage

```
dvoigt(y, mu = 0, sigma = 1, gamma = 1, fast= TRUE)
rvoigt(n = 1, mu = 0, sigma = 1, gamma = 1)
```

Arguments

n	The size of the random sample.
y	Vector of quantiles.
mu	The median of the Voigt.
sigma	Voigt first scale parameter.
gamma	Voigt second scale parameter.
fast	If TRUE (default) the Voigt density is calculated via Kendall's expression; if FALSE via numerical integration (slower but extended tail coverage).

Details

The Voigt distribution is the convolution of a Normal and a Cauchy. The density function is

$$f(y) = \frac{\Re(w(z))}{\sigma\sqrt{2\pi}}, \quad \sigma, \gamma > 0$$

where $w(z) = e^{-z^2} erfc(-iz)$ is the *Faddeeva* function, $z = \frac{y-\mu+i\gamma}{\sigma\sqrt{2}}$, mu is the median, sigma is the standard deviation of the normal component and gamma is the scale parameter of the Cauchy component. If mu, sigma and gamma are not specified they assume the default values of 0, 1 and 1, respectively.

Value

dvoigt gives the density and rvoigt generates random deviates.

Note

rvoigt is a wrapper of both rnorm and rcauchy functions. dvoigt uses integrate if fast=FALSE, see integrate; otherwise uses erf, see erf

Author(s)

Massimo Cannas [aut, cre], Nicola Piras [aut], Daniele Chiriu [ctb]

References

Kendall, D. G. (1938). The effect of radiation damping and Doppler broadening on the atomic absorption coefficient. *Zeitschrift fur Astrophysik, Vol. 16, p.308* https://adsabs.harvard.edu/full/1938ZA.....16..308K

Abramowitz, M., and I. A. Stegun (1972). Handbook of Mathematical Functions (with Formulas, Graphs, and Mathematical Tables). Dover, New York. https://personal.math.ubc.ca/~cbm/aands/notes.htm

See Also

evoigt

Examples

```
dvoigt(0)
# 0.2087093
# plot voigt and gaussian density
x = seq(from=-4,to=4, by=0.01)
plot(x, dvoigt(x), type="l",ylab="",ylim=c(0,0.8))
lines(x,dvoigt(x,0,1/3,1/3),col="blue")
lines(x,dnorm(x),lty=2, col="red")
mtext("dvoigt(x,0,1,1)", adj = 0,col="black")
mtext("dnorm(x)", adj = 1/2,col="red")
mtext("dvoigt(x,0,1/3,1/3)", adj = 1,col="blue")

# compare true and empirical density
rvoigt(1)
x = rvoigt(n=500)
q=quantile(x,0.99)
hist(x[x>-q & x<q], prob=TRUE, breaks=30,ylim=c(0,1.1*dvoigt(0)),main="", xlab="x")
x.grid = seq(-q,q,by=.1)
lines(x.grid, dvoigt(x.grid), type="l", ylab="", xlab="",col="red")
# compare with cauchy and normal density
par(mfrow=c(1,2))
x = rvoigt(n=500, mu = 0,sigma = 1,gamma = 0.1)
q=quantile(x,0.99)
hist(x[x>-q & x<q], prob=TRUE,breaks=20,col="lightgreen",main = "dvoigt(0, 1, 0.1)",xlab="x")
curve(dnorm(x),lty=1, add=TRUE)

y = rvoigt(n=500, mu = 0,sigma = 0.1,gamma = 1)
q=quantile(y,0.99)
hist(y[y>-q & y<q], prob=TRUE,breaks=25,col="lightblue",main = "dvoigt(0, 0.1, 1)",xlab="x")
curve(dcauchy(x),lty=1, add=TRUE)
dev.off()
```

Index