Package 'AcceptReject'

May 7, 2024

Title Acceptance-Rejection Method for Generating Pseudo-Random Observations

Version 0.1.2

Description Provides a function that implements the acceptance-rejection method in an optimized manner to generate pseudo-random observations for discrete or continuous random variables. Proposed by von Neumann J. (1951), <https://mcnp.lanl.gov/pdf_files/>, the function is optimized to work in parallel on Unix-based operating systems and performs well on Windows systems. The acceptance-rejection method implemented optimizes the probability of generating observations from the desired random variable, by simply providing the probability function or probability density function, in the discrete and continuous cases, respectively. Implementation is based on references CASELLA, George at al. (2004) <https://www.jstor.org/stable/4356322>, NEAL, Radford M. (2003) <https://www.jstor.org/stable/3448413> and Bishop, Christopher M. (2006, ISBN: 978-0387310732).

License GPL (>= 3)

Encoding UTF-8

URL https://prdm0.github.io/AcceptReject/

BugReports https://github.com/prdm0/AcceptReject/issues/

RoxygenNote 7.3.1

VignetteBuilder knitr

Imports assertthat, cli, ggplot2, glue, numDeriv, purrr, Rcpp, rlang, scales, scattermore

Suggests knitr, rmarkdown, cowplot, tictoc, testthat (>= 3.0.0)

Config/testthat/edition 3

LinkingTo Rcpp, RcppArmadillo

NeedsCompilation yes

Author Pedro Rafael D. Marinho [aut, cre] (<https://orcid.org/0000-0003-1591-8300>), Vera Lucia Damasceno Tomazella [ctb] (<https://orcid.org/0000-0002-6780-2089>)

Maintainer Pedro Rafael D. Marinho <pedro.rafael.marinho@gmail.com>

Repository CRAN

Date/Publication 2024-05-07 07:30:12 UTC

R topics documented:

`accept_reject` *Acceptance-Rejection Method*

Description

This function implements the acceptance-rejection method for generating random numbers from a given probability density function (pdf).

Usage

```
accept_reject(
  n = 1L,
  continuous = TRUE,
  f = NULL,
  args_f = NULL,
  f_base = NULL,
  random_base = NULL,
  args_f_base = NULL,
  xlim = NULL,
  c = NULL,
  parallel = FALSE,
  cores = NULL,
  warning = TRUE,
  ...
)
```

Arguments

`n`	The number of random numbers to generate.
`continuous`	A logical value indicating whether the pdf is continuous or discrete. Default is `TRUE`.
`f`	The probability density function (`continuous = TRUE`), in the continuous case or the probability mass function, in the discrete case (`continuous = FALSE`).

`args_f`	A list of arguments to be passed to the `f` function. It refers to the list of arguments of the target distribution.
`f_base`	Base probability density function (for continuous case).If `f_base = NULL`, a uniform distribution will be used. In the discrete case, this argument is ignored, and a uniform probability mass function will be used as the base.
`random_base`	Random number generation function for the base distribution passed as an argument to `f_base`. If `random_base = NULL` (default), the uniform generator will be used. In the discrete case, this argument is disregarded, and the uniform random number generator function will be used.
`args_f_base`	A list of arguments for the base distribution. This refers to the list of arguments that will be passed to the function `f_base`. It will be disregarded in the discrete case.
`xlim`	A vector specifying the range of values for the random numbers in the form `c(min, max)`. Default is `c(0, 100)`.
`c`	A constant value used in the acceptance-rejection method. If `NULL`, `c` will be estimated automatically.
`parallel`	A logical value indicating whether to use parallel processing for generating random numbers. Default is `FALSE`.
`cores`	The number of cores to be used in parallel processing. Default is `NULL`, i.e, all available cores.
`warning`	A logical value indicating whether to show warnings. Default is `TRUE`.
`...`	Additional arguments to be passed to the `optimize()`. With this argument, it is possible to change the tol argument of `optimize()`. Default is `tol = .Machine$double.eps^0.25)`.

Details

In situations where we cannot use the inversion method (situations where it is not possible to obtain the quantile function) and we do not know a transformation that involves a random variable from which we can generate observations, we can use the acceptance and rejection method. Suppose that X and Y are random variables with probability density function (pdf) or probability function (pf) f and g, respectively. In addition, suppose that there is a constant c such that

$$f(x) \leq c \cdot g(x), \quad \forall x \in \mathbb{R}.$$

for all values of t, with $f(t) > 0$. To use the acceptance and rejection method to generate observations from the random variable X, using the algorithm below, first find a random variable Y with pdf or pf g, that satisfies the above condition.

Algorithm of the Acceptance and Rejection Method:

1 - Generate an observation y from a random variable Y with pdf/pf g;

2 - Generate an observation u from a random variable $U \sim \mathcal{U}(0, 1)$;

3 - If $u < \frac{f(y)}{cg(y)}$ accept $x = y$; otherwise reject y as an observation of the random variable X and return to step 1.

Proof: Let's consider the discrete case, that is, X and Y are random variables with pf's f and g, respectively. By step 3 of the above algorithm, we have that $accept = x = y = u < \frac{f(y)}{cg(y)}$. That is,

$P(accept|Y = y) = \frac{P(accept \cap Y = y)}{g(y)} = \frac{P(U \leq f(y)/cg(y)) \times g(y)}{g(y)} = \frac{f(y)}{cg(y)}.$
Hence, by the Total Probability Theorem, we have that:

$P(accept) = \sum_y P(accept|Y = y) \times P(Y = y) = \sum_y \frac{f(y)}{cg(y)} \times g(y) = \frac{1}{c}.$

Therefore, by the acceptance and rejection method we accept the occurrence of $Y$ as being an occurrence of X with probability $1/c$. In addition, by Bayes' Theorem, we have that

$P(Y = y|accept) = \frac{P(accept|Y = y) \times g(y)}{P(accept)} = \frac{[f(y)/cg(y)] \times g(y)}{1/c} = f(y).$

The result above shows that accepting $x = y$ by the procedure of the algorithm is equivalent to accepting a value from X that has pf f.

The argument `c = NULL` is the default. Thus, the function `accept_reject()` estimates the value of `c` using the optimization algorithm `optimize()` using the Brent method. For more details, see `optimize()` function. If a value of `c` is provided, the function `accept_reject()` will use this value to generate the random observations. An inappropriate choice of `c` can lead to low efficiency of the acceptance and rejection method.

In Unix-based operating systems, the function `accept_reject()` can be executed in parallel. To do this, simply set the argument `parallel = TRUE`. The function `accept_reject()` utilizes the `parallel::mclapply()` function to execute the acceptance and rejection method in parallel. On Windows operating systems, the code will not be parallelized even if `parallel = TRUE` is set.

For the continuous case, a base density function can be used, where the arguments `f_base`, `random_base` and `args_f_base` need to be passed. If at least one of them is `NULL`, the function will assume a uniform density function over the interval `xlim`.

For the discrete case, the arguments `f_base`, `random_base` and `args_f_base` should be `NULL`, and if they are passed, they will be disregarded, as for the discrete case, the discrete uniform distribution will always be considered as the base. Sampling from the discrete uniform distribution has shown good performance for the discrete case.

The advantage of using parallelism in Unix-based systems is relative and should be tested for each case. Significant improvement is observed when considering parallelism for very large values of n. It is advisable to conduct benchmarking studies to evaluate the efficiency of parallelism in a practical situation.

Value

A vector of random numbers generated using the acceptance-rejection method. The return is an object of `class accept_reject`, but it can be treated as an atomic vector.

References

BISHOP, Christopher. 11.4: Slice sampling. Pattern Recognition and Machine Learning. Springer, 2006.

Brent, R. (1973) Algorithms for Minimization without Derivatives. Englewood Cliffs N.J.: Prentice-Hall.

CASELLA, George; ROBERT, Christian P.; WELLS, Martin T. Generalized accept-reject sampling schemes. Lecture Notes-Monograph Series, p. 342-347, 2004.

NEUMANN V (1951). "Various techniques used in connection with random digits." Notes by GE Forsythe, pp. 36–38.

NEAL, Radford M. Slice sampling. The Annals of Statistics, v. 31, n. 3, p. 705-767, 2003.

See Also

`inspect()`, `plot.accept_reject()`, `qqplot.accept_reject()`, `parallel::mclapply()` and `optimize()`.

Examples

```
set.seed(0) # setting a seed for reproducibility

x <- accept_reject(
  n = 2000L,
  f = dbinom,
  continuous = FALSE,
  args_f = list(size = 5, prob = 0.5),
  xlim = c(0, 5)
)
plot(x)

y <- accept_reject(
  n = 1000L,
  f = dnorm,
  continuous = TRUE,
  args_f = list(mean = 0, sd = 1),
  xlim = c(-4, 4)
)
plot(y)
```

`inspect` *Inspecting the theoretical density with the base density*

Description

Inspect the probability density function used as the base with the theoretical density function from which observations are desired.

Usage

```
inspect(
  f,
  args_f,
  f_base,
  args_f_base,
  xlim,
  c = 1,
  alpha = 0.4,
  color_intersection = "#BB9FC9",
  color_f = "#F890C2",
  color_f_base = "#7BBDB3"
)
```

Arguments

`f`	Theoretical density function.
`args_f`	List of arguments for the theoretical density function.
`f_base`	Base density function.
`args_f_base`	List of arguments for the base density function.
`xlim`	The range of the x-axis.
`c`	A constant that covers the base density function, with $c \geq 1$. The default value is 1.
`alpha`	The transparency of the base density function. The default value is 0.4
`color_intersection`	Color of the intersection between the base density function and theoretical density functions.
`color_f`	Color of the base density function.
`color_f_base`	Color of the theoretical density function.

Details

The function `inspect()` returns an object of the gg and `ggplot` class that compares the probability density of two functions and is not useful for the discrete case, only for the continuous one. Finding the parameters of the base distribution that best approximate the theoretical distribution and the smallest value of `c` that can cover the base distribution is a great strategy. Something important to note is that the plot provides the value of the area of intersection between the theoretical probability density function we want to generate observations from and the probability density function used as the base. It's desirable for this value to be as close to 1 as possible, ideally

1. When the intersection area between the probability density functions is 1, it means that the base probability density function passed to the `f_base` argument overlaps the theoretical density function passed to the `f` argument. This is crucial in the acceptance-rejection method. However, even if you don't use the `inspect()` function to find a suitable distribution, by finding viable args_base (list of arguments passed to f_base) and the value of `c` so that the intersection area is 1, the `accept_reject()` function already does this for you. The `inspect()` function is helpful for finding a suitable base distribution, which increases the probability of acceptance, further reducing computational cost. Therefore, inspecting is a good practice.

If you use the `accept_reject()` function, even with parallelism enabled by specifying `parallel = TRUE` in `accept_reject()` and find that the generation time is high for your needs, consider inspecting the base distribution.

Value

An object of the gg and `ggplot` class comparing the theoretical density function with the base density function. The object shows the compared density functions, the intersection area between them, and the value of the area.

See Also

`accept_reject()`, `print.accept_reject()` and `plot.accept_reject()`.

Examples

```
# Considering c = 1 (default)
inspect(
   f = dweibull,
   f_base = dgamma,
   xlim = c(0,5),
   args_f = list(shape = 2, scale = 1),
   args_f_base = list(shape = 2.1, rate = 2),
   c = 1
)

# Considering c = 1.35.
inspect(
   f = dweibull,
   f_base = dgamma,
   xlim = c(0,5),
   args_f = list(shape = 2, scale = 1),
   args_f_base = list(shape = 2.1, rate = 2),
   c = 1.35
)

# Plotting f equal to f_base. This would be the best-case scenario, which,
# in practice, is unlikely.
inspect(
   f = dgamma,
   f_base = dgamma,
   xlim = c(0,5),
   args_f = list(shape = 2.1, rate = 2),
   args_f_base = list(shape = 2.1, rate = 2),
   c = 1
)
```

`plot.accept_reject` *Plot Accept-Reject*

Description

Inspects the probability function (discrete case) or probability density (continuous case) by comparing the theoretical case with the observed one.

Usage

```
## S3 method for class 'accept_reject'
plot(
  x,
  color_observed_density = "#BB9FC9",
  color_true_density = "#F890C2",
  color_bar = "#BB9FC9",
  color_observable_point = "#7BBDB3",
  color_real_point = "#F890C2",
  alpha = 0.3,
  hist = TRUE,
  ...
)
```

Arguments

`x`	An object of class `accept reject`
`color_observed_density`	Observed density color (continuous case).
`color_true_density`	True histogram density color (continuous case)
`color_bar`	Bar chart fill color (discrete case)
`color_observable_point`	Color of generated points (discrete case)
`color_real_point`	Color of real probability points (discrete case)
`alpha`	Bar chart transparency (discrete case) and observed density (continuous case)
`hist`	If `TRUE`, a histogram will be plotted in the continuous case, comparing the theoretical density with the observed one. If `FALSE`, `ggplot2::geom_density()` will be used instead of the histogram.
`...`	Additional arguments.

Details

The function `plot.accept_reject()` is responsible for plotting the probability function (in the discrete case) or the probability density (in the continuous case), comparing the theoretical case with the observed one. It is useful, therefore, for inspecting the quality of the samples generated by the acceptance-rejection method. The returned plot is an object of classes gg and `ggplot`. Easily, you can further customize the plot.

The function `plot.accept_reject()`, or simply `plot()`, constructs the plot for inspection and expects an object of class `accept_reject` as an argument.

Value

An object of class gg and `ggplot` from the package **ggplot2**. The function `plot.accept_reject()` expects an object of class `accept_reject` as an argument.

See Also

`accept_reject()` and `print.accept_reject()`.

Examples

```
x <- accept_reject(
   n = 1000L,
   f = dbinom,
   continuous = FALSE,
   args_f = list(size = 10, prob = 0.5),
   xlim = c(0, 10)
)
plot(x)

y <- accept_reject(
  n = 500L,
  f = dnorm,
  continuous = TRUE,
  args_f = list(mean = 0, sd = 1),
  xlim = c(-4, 4)
)
plot(y)
```

`print.accept_reject` *Print method for accept_reject objects*

Description

Print method for accept_reject objects.

Usage

```
## S3 method for class 'accept_reject'
print(x, n_min = 10L, ...)
```

Arguments

`x`	An accept_reject object.
`n_min`	Minimum number of observations to print.
`...`	Additional arguments.

Details

The function `print.accept_reject()` is responsible for printing an object of class `accept_reject` in a formatted manner, providing some information about the `accept_reject` object, including the number of observations, the value of the constant c that maximizes acceptance, and the acceptance probability $1/c$. Additionally, it prints the first generated observations. The function `print.accept_reject()` delivers formatted output when executing an object of class `accept_reject` in the console or when executing the function `print()` on an object of class `accept_reject`, returned by the function `accept_reject()`.

Value

An object of class `character`, providing a formatted output with some information about the `accept_reject` object, including the number of observations, the value of the constant c that maximizes acceptance, and the acceptance probability $1/c$. Additionally, it prints the first generated observations. The function `print.accept_reject()` enables formatting when executing an object of class 'accept_reject' in the console or when executing the function `print()` on an object of class `accept_reject`, returned by the function `accept_reject()`.

See Also

`accept_reject()` and `plot.accept_reject()`.

Examples

```
set.seed(0) # setting a seed for reproducibility
x = accept_reject(
   n = 2000L,
   f = dbinom,
   continuous = FALSE,
   args_f = list(size = 10, prob = 0.5),
   xlim = c(0, 10)
)
print(x)
```

`qqplot` *QQ-Plot QQ-Plot between observed quantiles and theoretical quantiles.*

Description

QQ-Plot QQ-Plot between observed quantiles and theoretical quantiles.

Usage

```
qqplot(x, ...)
```

Arguments

`x`	Object of the class `accept_reject` returned by the function `accept_reject()`.
`...`	Additional arguments to be passed to methods.

Details

Generic method to plot the QQ-Plot between observed quantiles and theoretical quantiles. The generic method will call the specific method `qqplot.accept_reject()`, which operates on objects of class accept_reject returned by the function `accept_reject()`.

Value

An object of classes gg and `ggplot` with the QQ-Plot of theoretical quantiles versus observed quantiles.

See Also

`accept_reject()`, `print.accept_reject()`, `plot.accept_reject()` and `inspect()`.

`qqplot.accept_reject` *QQ-Plot Plot the QQ-Plot between observed quantiles and theoretical quantiles.*

Description

QQ-Plot Plot the QQ-Plot between observed quantiles and theoretical quantiles.

Usage

```
## S3 method for class 'accept_reject'
qqplot(
  x,
  alpha = 0.5,
  color_points = "#F890C2",
  color_line = "#BB9FC9",
  size_points = 1,
  size_line = 1,
  ...
)
```

Arguments

`x`	Object of the class accept_reject returned by the function `accept_reject()`.
`alpha`	Transparency of the points and reference line representing where the quantiles should be (theoretical quantiles).
`color_points`	Color of the points (default is `"#F890C2"`).
`color_line`	Color of the reference line (detault is `"#BB9FC9"`).
`size_points`	Size of the points (default is `1`).
`size_line`	Thickness of the reference line (default is `1`).
`...`	Additional arguments for the `quantile()` function. For instance, it's possible to change the algorithm type for quantile calculation.

Details

The function `qqplot.accept_reject()` for samples larger than ten thousand, the `geom_scattermost()` function from the **scattermore** library is used to plot the points, as it is more efficient than `geom_point()` from the **ggplot2** library.

Value

An object of classes gg and ggplot with the QQ-Plot between the observed quantiles generated by the return of the function accept_reject() and the theoretical quantiles of the true distribution.

See Also

qqplot.accept_reject(), accept_reject(), plot.accept_reject(), inspect() and qqplot().

Examples

```
set.seed(0) # setting a seed for reproducibility

x <- accept_reject(
  n = 2000L,
  f = dbinom,
  continuous = FALSE,
  args_f = list(size = 5, prob = 0.5),
  xlim = c(0, 5)
)
qqplot(x)

y <- accept_reject(
  n = 1000L,
  f = dnorm,
  continuous = TRUE,
  args_f = list(mean = 0, sd = 1),
  xlim = c(-4, 4)
)
qqplot(y)
```

Index