

Package: ipfp (via r-universe)

October 11, 2026

Type Package

Title Fast Implementation of the Iterative Proportional Fitting
Procedure in C

Version 1.0.2

Author Alexander W Blocker

Maintainer Alexander W Blocker <ablocker@gmail.com>

Description A fast (C) implementation of the iterative proportional
fitting procedure.

License Apache License (== 2.0)

LazyLoad yes

URL <https://github.com/awblocker/ipfp>

RoxygenNote 7.1.2

Repository <https://awblocker.r-universe.dev>

Date/Publication 2022-05-04 18:33:06 UTC

RemoteUrl <https://github.com/awblocker/ipfp>

RemoteRef HEAD

RemoteSha 2100016c76f8d85a1bde4fd479ffb8ed427a37af

Contents

ipfp	2
Index	3

ipfp

*Function to run IPFP (iterative proportional fitting procedure)***Description**

Use IPFP starting from x_0 to produce vector x s.t. $Ax = y$ within tolerance. Need to ensure that $x_0 > 0$.

Usage

```
ipfp(
  y,
  A,
  x0,
  tol = sqrt(.Machine$double.eps),
  maxit = 1000,
  verbose = FALSE,
  full = FALSE
)
```

Arguments

<code>y</code>	numeric constraint vector (length nrow)
<code>A</code>	constraint matrix (nrow x ncol)
<code>x0</code>	numeric initial vector (length ncol)
<code>tol</code>	numeric tolerance for IPFP; defaults to <code>sqrt(.Machine\$double.eps)</code>
<code>maxit</code>	integer maximum number of iterations for IPFP; defaults to <code>1e3</code>
<code>verbose</code>	logical parameter to select verbose output from C function
<code>full</code>	logical parameter to select full return (with diagnostic info)

Value

if not full, a vector of length ncol containing solution obtained by IPFP. If full, a list containing solution (as `x`), the number of iterations (as `iter`), and the L2 norm of $Ax - y$ (as `errNorm`)

Examples

```
A <- matrix(c(1,0,0, 1,0,0, 0,1,0, 0,1,0, 0,0,1), nrow=3)
x <- rgamma(ncol(A), 10, 1/100)
y <- A %*% x
x0 <- x * rgamma(length(x), 10, 10)
ans <- ipfp(y, A, x0, full=TRUE)
print(ans)
print(x)
```

Index

- * **array**
 - ipfp, [2](#)
- * **iteration**
 - ipfp, [2](#)

ipfp, [2](#)