

Package: sfit (via r-universe)

September 2, 2024

Version 0.3.2

Depends R (>= 2.3.0)

Imports R.methodsS3 (>= 1.8.1), R.oo (>= 1.24.0)

Title Multidimensional Simplex Fitting

Author Henrik Bengtsson [aut, cre, cph], Pratyaksha Wirapati [aut, cph]

Maintainer Henrik Bengtsson <henrikb@braju.com>

Description Methods for robustly fitting a K-dimensional simplex in M dimensions.

License LGPL (>= 2.1)

LazyLoad yes

ByteCompile TRUE

URL <https://github.com/HenrikBengtsson/sfit>

BugReports <https://github.com/HenrikBengtsson/sfit/issues>

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

RemoteUrl <https://github.com/HenrikBengtsson/sfit>

RemoteRef master

RemoteSha 922cc89d4232cce3625670abe44e2c186b4e4167

Contents

sfit-package	2
cfit	3

Index	6
--------------	----------

sfit-package

Package sfit

Description

Methods for robustly fitting a K-dimensional simplex in M dimensions.

Installation and updates

To install this package, see <http://www.braju.com/R/>.

To get started

To get started, see:

1. `cfit()` - To fit a K-dimensional simplex in an N-dimensional space.

How to cite this package

Please cite [1] and [2] below.

Wishlist

The current interface from R is very ad hoc; it dumps all data to file(s), calls 'cfit' with the correct parameters via `pipe()` (see [connections](#)) and parses the output files from 'cfit'. Ideally, we would link the cfit code to R via `.Call()` (see [Foreign](#)), but that is for the future. The current solutions has been verified to work on Windows XP, Linux and OSX.

License

The releases of this package is licensed under LGPL version 2.1 or newer.

The development code of the packages is under a private licence (where applicable) and patches sent to the author fall under the latter license, but will be, if incorporated, released under the "release" license above.

Author(s)

The algorithm and its C source code implementation is work of Pratyaksha Wirapati. The R wrapper is work of Henrik Bengtsson.

References

- [1] P. Wirapati, & T. Speed, *Fitting polyhedral cones and simplices to multivariate data points*, Walter and Eliza Hall Institute of Medical Research, December 30, 2001.
- [2] P. Wirapati and T. Speed, *An algorithm to fit a simplex to a set of multidimensional points*, Walter and Eliza Hall Institute of Medical Research, January 15, 2002.

cfit	<i>Fits a K-dimensional simplex in M dimensions</i>
------	---

Description

Fits a K-dimensional simplex in M dimensions. A K-dimensional simplex is the K-dimensional generalization of a triangle.

Usage

```
## S3 method for class 'matrix'
cfit(y, k=ncol(y) + 1, dump=1, chopless=NULL, chopmore=NULL, maxiter=NULL, ...,
     retX=FALSE, cfit=getOption("cfit"), verbose=FALSE)
```

Arguments

y	Matrix or data frame of size IxN containing I rows of vectors in R^N .
k	The number of vertices of the fitted simplex. By default, the number of vertices is equal to the number of dimension (N) + 1.
dump	The output format.
chopless, chopmore	Lower and upper percentile thresholds at which extreme data points are assigned zero weights.
maxiter	"maximum number of REX steps". Default value is 60.
...	Named argument passed to the external 'cfit' program.
retX	If TRUE , an estimate of X is returned, otherwise not.
cfit	Shell command to call the 'cfit' executable.
verbose	If TRUE , verbose output is displayed, otherwise not.

Details

Let $Y = (y_1, \dots, y_I)$ where $y_i = (y_{i1}, \dots, y_{iN})$ is an observation in N dimensions. Let $M = (\mu_1, \dots, \mu_K)$ be the K -dimensional simplex where μ_k is a vertex in N dimensions. Let $X = (x_1, \dots, x_I)$ where $x_i = (x_{i1}, \dots, x_{iN})$. The simplex fitting algorithm decompose Y into:

$$Y \approx MX$$

such that $\sum_i x_{ik} = 1$.

Value

Returns a named **list** structure elements:

M	IxN matrix where each rows is the coordinate for one of the vertices.
X	(optional) the IxN matrix X.

Author(s)

Algorithm and C code/binary by Pratyaksha J. Wirapati. R wrapper by Henrik Bengtsson.

References

- [1] P. Wirapati, & T. Speed, *Fitting polyhedral cones and simplices to multivariate data points*, Walter and Eliza Hall Institute of Medical Research, December 30, 2001.
 [2] P. Wirapati and T. Speed, *An algorithm to fit a simplex to a set of multidimensional points*, Walter and Eliza Hall Institute of Medical Research, January 15, 2002.

Examples

```
# - - - - -
# Simulate data
# - - - - -
N <- 1000

# Simulate genotypes
g <- sample(c("AA", "AB", "AB", "BB"), size=N, replace=TRUE)

# Simulate concentrations of allele A and allele B
X <- matrix(rexp(N), nrow=N, ncol=2)
colnames(X) <- c("A", "B")
X[g == "AA", "B"] <- 0
X[g == "BB", "A"] <- 0
X[g == "AB",] <- X[g == "AB",] / 2

# Transform noisy X
xi <- matrix(rnorm(2*N, mean=0, sd=0.05), ncol=2)
a0 <- c(0,0)+0.3
A <- matrix(c(0.9, 0.1, 0.1, 0.8), nrow=2, byrow=TRUE)
A <- apply(A, MARGIN=2, FUN=function(u) u / sqrt(sum(u^2)))
Z <- t(a0 + A %*% t(X + xi))

# Add noise to Y
eps <- matrix(rnorm(2*N, mean=0, sd=0.05), ncol=2)
Y <- Z + eps

layout(matrix(1:4, ncol=2, byrow=TRUE))
par(mar=c(5,4,3,2)+0.1)
xlab <- "Allele A"
ylab <- "Allele B"
lim <- c(-0.5,8)
plot(X, xlab=xlab, ylab=ylab, xlim=lim, ylim=lim)
points(Z, col="blue")
points(Y, col="red")

legend("topright", pch=19, pt.cex=2, legend=c("X", "Z", "Y"),
      col=c("black", "blue", "red"), title="Variables:", bg="#eeeeee")
```

```

# - - - - -
# Fit model
# - - - - -
alpha <- c(0.10, 0.075, 0.05, 0.03, 0.01, 0.001)
fit <- cfit(Y, dump=2, alpha=alpha, q=2, Q=98)
Ms <- fit$M
col <- terrain.colors(length(Ms))
col[length(Ms)] <- "red"

plot(Y, cex=0.8, xlab=xlab, ylab=ylab, xlim=lim, ylim=lim, main="Y")

for (kk in seq_along(Ms)) {
  M <- Ms[[kk]]
  points(M, pch=19, cex=2.5, col=col[kk])
  lines(M, col=col[kk], lwd=2)
  text(M, cex=0.8, labels=kk)
}

legend("topright", pch=19, pt.cex=2, legend=c(alpha, "final"),
       col=col, title=expression(alpha), bg="#eeeeee")

apex <- which.min(apply(M, MARGIN=1, FUN=function(u) sum(u^2)))
a0hat <- M[apex,]
Ahat <- M[-apex,]
Ahat <- apply(Ahat, MARGIN=2, FUN=function(u) u / sqrt(sum(u^2)))
if (sum(Ahat[c(1,4)]^2) < sum(Ahat[c(2,3)]^2)) {
  Ahat <- matrix(Ahat[c(2,1,4,3)], nrow=2)
}
Ainv <- solve(Ahat)
Xhat <- t(Ainv %*% (t(Y) - a0hat))

cat("True A:\n")
print(A)

cat("Estimated A:\n")
print(Ahat)

plot(Xhat, cex=0.8, xlab=xlab, ylab=ylab, xlim=lim, ylim=lim, main=expression(hat(X)))
x1 <- par("usr")[2]
y1 <- par("usr")[4]
lines(x=c(0,x1), y=c(0,0), col="red", lwd=2)
lines(x=c(0,0), y=c(0,y1), col="red", lwd=2)
lines(x=c(0,x1), y=c(0,y1), col="blue", lwd=2)

plot(X[,1], Xhat[,1], cex=0.8, xlab=expression(X), ylab=expression(hat(X)), xlim=lim, ylim=lim)
points(X[,2], Xhat[,2], cex=0.8, col="red")
abline(a=0, b=1, lwd=2)

```

Index

*** methods**

 cfit, [3](#)

*** package**

 sfit-package, [2](#)

cfit, [2](#), [3](#)

connections, [2](#)

Foreign, [2](#)

list, [3](#)

matrix, [3](#)

sfit (sfit-package), [2](#)

sfit-package, [2](#)

TRUE, [3](#)