Package 'kza'

February 14, 2012

Version 2.01

Date 2011/08/24

Title Kolmogorov-Zurbenko Adaptive Filters

Author Brian Close <brian.close@gmail.com> and Igor Zurbenko <IZurbenko@albany.edu>

Maintainer Brian Close <brian.close@gmail.com>

Description Time Series Analysis including break detection, spectral analysis, KZ Fourier Transforms.

Depends polynom

SystemRequirements fftw (>= 3.2.2)

LazyLoad yes

License GPL-3

Repository CRAN

Date/Publication 2011-09-08 04:11:06

R topics documented:

`fars` *Fatal Analysis Reporting System*

Description

Highway fatality data from the National Highway Safety and Traffic Administration.

Usage

```
fars
```

Format

A vector containing over 5000 observations.

Source

National Highway Traffic Safety Administration

`kz` *Kolmogorov-Zurbenko filter*

Description

Kolmogorov-Zurbenko low-pass linear filter.

Usage

```
kz(x, m, k = 3)
```

Arguments

`x`	A vector of the time series.
`m`	Window size for the filter.
`k`	Number of iterations.

Details

KZ is an iterated moving average. The filter can be used with missing values.

References

Zurbenko, I. G., 1986: The spectral Analysis of Time Series. North-Holland, 248 pp.

Examples

```
#seperation of signals
yrs <- 20
t <- seq(0,yrs,length=yrs*365)
y <- sin(2*pi*t) + sin(3*pi*t)
k.kz <- kz(y,365/2)

par(mfrow=c(3,1))
plot(y,type="l",main="y = sin(2*pi*t)+sin(3*pi*t)")
plot(k.kz,type="l",main="KZ filter")

r <- y - 4*k.kz
plot(r,type="l",main="(y - 4*kz) ~ sin(3*pi*t)")

#another example
#remove noise and high frequency
yrs <- 20
t <- seq(0,yrs,length=yrs*365)
set.seed(6); e <- rnorm(n = length(t), sd = 1.0)
y <- sin(2*pi*t) + sin(3*pi*t) + e

k.kz <- kz(y,365/4)

par(mfrow=c(2,1))
plot(y,type="l",main="y = sin(2*pi*t)+sin(3*pi*t) + noise")
plot(k.kz,type="l",main="KZ filter")
```

---

kza *Kolmogorov-Zurbenko Adaptive*

Description

KZA will recover 2-dimensional or 3-dimensional image or signal buried in noise.

Usage

```
kza(x, m, y = NULL, k = 3, min_size = round(0.05*m), tol = 1.0e-5, impute_tails = FALSE)
```

Arguments

x	A vector of the time series or a matrix (2d) or an array (3d) of an image.
m	The window for the filter.
y	The filtered output from kz.
k	The number of iterations.
min_size	Minimum size of window q.
tol	The smallest value to accept as nonzero.
impute_tails	The default is to drop the tails.

Details

The selection of parameters of KZA depend on the nature of the data. This function can take a long time to run, depending on the number of dimensions and the size of the dimensions.

Author(s)

Brian Close <brian.close@gmail.com> and Igor Zurbenko <IZurbenko@albany.edu>

References

I. Zurbenko, P.S. Porter, S.T. Rao, J.Y. Ku, R. Gui, R.E. Eskridge Detecting Discontinuities in Time Series of Upper-air Data: Development and Demonstration of an Adaptive Filter Technique. Journal of Climate: (1996) Vol. 9, No. 12, pp. 3548 3560. http://ams.allenpress.com/amsonline/?request=get-abstract&issn=1520-0442&volume=009&issue=12&page=3548

Kevin L. Civerolo, Elvira Brankov, S. T. Rao, Igor Zurbenko Assessing the impact of the acid deposition control program. Atmospheric Environment 35 (2001) 4135-4148 http://www.elsevier.com/locate/atmosenv

J.Chen, I.Zurbenko, Nonparametric Boundary detection, Communications in Statistics, Theory and Methods, Vol.26, 12, 2999-3014, 1997.

Examples

```
#######
# this is an example of detection of a break point in a time series
#######
yrs <- 20
t <- seq(0,yrs,length=yrs*365)
m <- 365

#noise
e <- rnorm(n = length(t),0,1)
trend <- seq(0,-1,length=length(t))

#signal
bkpt <- 3452
brk <- c(rep(0,bkpt),rep(0.5,length(t)-bkpt))
signal <- trend + brk

# y = seasonal + trend + break point + noise
y <- sin(2*pi*t) + signal + e

k.kz <- kz(y,m)

# kza reconstruction of the signal
k.kza <- kza(y,m,y=k.kz,min_size=10)

par(mfrow=c(2,1))
plot(y,type="l", ylim=c(-3,3))
plot(signal,type="l",ylim=c(-3,3),
    main="Signal and KZA Reconstruction")
```

```
lines(k.kza$kza, col=4)

######################
# image detection (2d)
######################
a <- matrix(rep(0,100*100),nrow=100)
a[35:70,35:70]<-1
a <- a + matrix(rnorm(100*100,0,1),nrow=100)
y<-kz(a,m=15,k=3)
v <- kza(a,m=15,y=y,k=3,impute_tails=TRUE)

x <- seq(1,100)
y <- x
op <- par(bg = "white")

###
#noise
###
persp(x, y, a, zlab="z", zlim=c(0,2), ticktype="detailed", theta = 30, phi = 30, col = "lightblue")

###
#kza filtered
###
X11()
persp(x, y, v$kza, zlab="z", zlim=c(0,2), ticktype="detailed", theta = 30, phi = 30, col = "lightblue")

###
# another view
###
par(mfrow=c(1,2))
image(a,col=gray(seq(0,1,1/255)))
image(v$kza,col=gray(seq(0,1,1/255)))
par(mfrow=c(1,1))
```

---

kzft *Kolmogorov-Zurbenko Fourier Transform*

Description

Kolmogorov-Zurbenko Fourier Transform is an iterated Fourier transform.

Usage

```
kzft(x, m = length(x), k = 1, f = NULL, dim = 1, index = NULL, trim=FALSE)
max_freq(x, m, start=1, t = NULL)
transfer_function(m,k,lamda=seq(-0.5,0.5,by=0.01),omega=0)
```

Arguments

x	The raw time series
m	The window size for transform
k	The number of iterations for applying the KZFT
f	The frequency that KZFT is applied at.
dim	A value of 1 will return a vector of the given frequency and a value of 2 will return a matrix (spectra).
index	An indexing set
trim	Remove k*m from edges of array.
start	Search for max amplitude begining at start.
t	An indexing set
lamda	The frequencies used for the calculating the transfer function.
omega	The frequency that KZFT is applied at.

Details

Kolmogorov-Zurbenko Fourier Transform (KZFT) is the Fourier transform applied over every segment of length m iterated k times. If a frequency is not selected then the frequency with the largest amplitude is used. The approach taken here is to iterate fft to create a matrix of frequencies over time, another approach is to use matrices for the KZFT transform (see the KZFT package by Wei Yang in R).

The frequency_kzft function returns the frequency with the largest variation.

References

I. G. Zurbenko, The spectral Analysis of Time Series. North-Holland, 1986.

I. G. Zurbenko, P. S. Porter, Construction of high-resolution wavelets, Signal Processing 65: 315-327, 1998.

R. Neagu, I. G. Zurbenko, Tracking and separating non-stationary multi-component chirp signals, J. Franklin Inst., 499-520, 2002.

R. H. Shumway, D. S. Stoffer, Time Series Analysis and Its Applications: With R Examples, Springer, 2006.

Wei Yang and Igor Zurbenko, kzft: Kolmogorov-Zurbenko Fourier Transform and Applications, R-Project 2007.

Igor G. Zurbenko, Amy L. Potrzeba, Tidal Waves in Atmosphere and Their Effects, Acta Geophysica Volume 58, Number 2, 356-373

See Also

kzp, kztp,

Examples

```
# example taken from Wei Yang's KZFT package
# coefficients of kzft(201,5)

# function to calculate polnomial coefficients for kzft
#require(polynom)
#coeff <- function(m, k)
#{
#    poly<-polynomial(rep(1,m))
#    polyk<-poly^k
#    coef<-as.vector(polyk)
#    coef<-coef/m^k
#    M=(m-1)*k+1
#    return(coef[1:M])
#}

#a<-coeff(201,5);
#t<-seq(1:1001)-501;
#z<-cos(2*pi*0.025*t);
#plot(z*a,type="l",xlab="Time", ylab="Coefficient", main="Coefficients of the kzft");
#lines(a);
#lines(-1*a);

# example taken from Wei Yang's KZFT package
# transfer function of the kzft(201,5) at frequency 0.025
lamda<-seq(-0.1,0.1,by=0.001)
tf1<-transfer_function(201,1,lamda,0.025)
tf2<-transfer_function(201,5,lamda,0.025)
matplot(lamda,cbind(log(tf1),log(tf2)),type="l",ylim=c(-15,0),
ylab="Natural log transformation of the coefficients",
xlab="Frequency (cycles/time unit)",
    main="Transfer function of kzft(201,5) at frequency 0.025")

# example with missing values
period=101
f<-1/period
t<-1:2000
s<-1*sin(2*pi*f*t)
x<-s
noise<-3*rnorm(length(t))
x<-s+noise
m=101

rand_idx <- sample(t,100,replace=FALSE)
x[rand_idx]<-NA
t[rand_idx]<-NA
x<-as.vector(na.omit(x))
t<-as.vector(na.omit(t))

system.time(z1<-kzft(x, m=m, k=1, f=f, dim=1, index=t))
system.time(z2<-kzft(x, m=m, k=2, f=f, dim=1, index=t))
system.time(z3<-kzft(x, m=m, k=3, f=f, dim=1, index=t))
```

```
par(mfrow=c(2,2))
plot(x,type="l",main="Original time series",xlab="t", ylab="y")
lines(s,col="blue")
plot(2*Re(z1),type="l",main="kzft(101,1)",xlab="t", ylab="y", ylim=c(-6,6))
lines(s,col="blue")
plot(2*Re(z2),type="l",main="kzft(101,2)",xlab="t", ylab="y", ylim=c(-6,6))
lines(s,col="blue")
plot(2*Re(z3),type="l",main="kzft(101,3)",xlab="t", ylab="y", ylim=c(-6,6))
lines(s,col="blue")
par(mfrow=c(1,1))
```

---

kzp *Kolmogorov-Zurbenko Periodogram*

Description

Kolmogorov-Zurbenko periodogram and smoothing using DiRienzo-Zurbenko (DZ).

Usage

```
kzp(y, m=length(y), k=3, index=NULL, double_frequency=FALSE)
## S3 method for class 'kzp'
smooth(object, log=TRUE, smooth_level=0.05, method = "DZ")
## S3 method for class 'kzp'
nonlinearity(x)
## S3 method for class 'kzp'
variation(x)
## S3 method for class 'kzp'
summary(object, digits=getOption("digits"), top=NULL, ...)
## S3 method for class 'kzp'
plot(x, ...)
```

Arguments

y	The raw data.
m	The width of filtering window
k	The number of iterations for the KZFT
index	An indexing set
double_frequency	The return vector is half the width of the filtering window, setting this to true will give the second half.
object	Output from kzp function.
log	Use logarithm values for smoothing.
smooth_level	Percentage of smoothness to apply.
method	Method used for smoothing; choices are "DZ" or "NZ".

`digits`	precision of output.
`top`	list top values
`...`	Other parameters.
`x`	periodogram

Details

The Kolmogorov-Zurbenko Periodogram is an estimate of the spectral density using the Kolmogorov-Zurbenko Fourier Transform (KZFT).

References

I. G. Zurbenko, 1986: The spectral Analysis of Time Series. North-Holland, 248 pp.

I. G. Zurbenko, P. S. Porter, Construction of high-resolution wavelets, Signal Processing 65: 315-327, 1998.

A. G. DiRienzo, I. G. Zurbenko, Semi-adaptive nonparametric spectral estimation, Journal of Computational and Graphical Statistics 8(1): 41-59, 1998.

R. Neagu, I. G. Zurbenko, Algorithm for adaptively smoothing the log-periodgram, Journal of the Franklin Institute 340: 103-123, 2003.

Wei Yang and Igor Zurbenko, kzft: Kolmogorov-Zurbenko Fourier Transform and Applications, R-Project 2007.

See Also

`kzft`, `kztp`,

Examples

```
t<-1:6000
f1<-0.03
f2<-0.04
noise<-15*rnorm(length(t))
amp=1.5
s<-amp*sin(2*pi*f1*t)+amp*sin(2*pi*f2*t)
system.time(a<-kzp(s+noise,500,k=3))
b<-smooth.kzp(a, smooth_level=0.08)
par(mfrow=c(3,1))
plot(periodogram(s+noise),type='l')
plot(a)
plot(b)
par(mfrow=c(1,1))

# signal/noise
signal<-kzft(s+noise,m=500,k=3,dim=1)
print(paste("signal-to-noise ratio = ", round(sqrt(var(2*Re(signal))/var(s+noise)),4) ))

summary(a, digits=2, top=2)
```

kzs — *Kolmogorov-Zurbenko Spline*

Description

Kolmogorov-Zurbenko Spline

Usage

```
kzs(y,m=NULL,k=3,t=NULL)
```

Arguments

`y`	data
`m`	smooth
`k`	The number of iterations for applying the KZFT
`t`	An indexing set

Details

Kolmogorov-Zurbenko Spline is essentially the Kolmogorov-Zurbenko Fourier Transform at the zero frequency.

References

I. G. Zurbenko, The spectral Analysis of Time Series. North-Holland, 1986.

I. G. Zurbenko, P. S. Porter, Construction of high-resolution wavelets, Signal Processing 65: 315-327, 1998.

R. H. Shumway, D. S. Stoffer, Time Series Analysis and Its Applications: With R Examples, Springer, 2006.

Derek Cyr and Igor Zurbenko, kzs: Kolmogorov-Zurbenko Spatial Smoothing and Applications, R-Project 2008.

Derek Cyr and Igor Zurbenko, A Spatial Spline Algorithm and an Application to Climate Waves Over the United States, 2008 Joint Statistical Meetings.

See Also

`kzft`,

Examples

```
n <- 1000
x <- (1:n)/n
true<-((exp(2.5*x)+sin(25*x))-1)/3

noise <- rnorm(n)
y <- true + noise

a<-kzs(y,m=60)

par(mfrow=c(2,1))
plot(y,type='l')
lines(true,col="red")

plot(a,type='l', ylim=c(-2,4))
lines(true,col="red")
par(mfrow=c(1,1))

################
# second example
################
t <- seq(from = -round(400*pi), to = round(400*pi), by = .25)
ts <- 0.5*sin(sqrt((2*pi*abs(t))/200))
signal <- ifelse(t < 0, -ts, ts)
et <- rnorm(length(t), mean = 0, sd = 1)
yt <- et + signal

b<-kzs(yt,m=400)
par(mfrow=c(2,1))
plot(yt,type='l')
lines(signal,col="red")

plot(b,type='l', ylim=c(-0.5,1))
lines(signal,col="red")
par(mfrow=c(1,1))
```

---

kzsv — *Kolmogorov-Zurbenko Adaptive filter with Sample Variance.*

Description

Sample variance of a Kolmogorov-Zurbenko adaptive filter. You want a sigma of at least 3.

Usage

```
kzsv(object)
```

Arguments

`object`	The resultant object from kza function.

Examples

```
x <- c(rep(0,4000),rep(0.5,2000),rep(0,4000))
noise <- rnorm(n = 10000, sd = 1.0) # normally-distributed random variates
v <- x + noise
a<-kza(v, m=1000, k=3)
sv<-kzsv(a)
```

kztp — *Kolmogorov-Zurbenko Third-Order Periodogram*

Description

Kolmogorov-Zurbenko Third-Order Periodogram for estimating spectrums

Usage

```
kztp(x,m,k,box=c(0,0.5,0,0.5))
```

Arguments

`x`	The signal.
`m`	The window size for the kzft filter.
`k`	The number of iterations.
`box`	The window for the application of third-order periodgram.

Details

The Kolmogorov-Zurbenko Third-Order Periodogram is used to estimate spectral density of a signal. The smoothing methods are adaptive allowing the bandwidth of the spectral window to vary according to the smoothness of the underlying spectral density. For details, please see to DiRienzo and Zurbenko (1998) and Neagu and Zurbenko (2003).

References

I. G. Zurbenko, 1986: The spectral Analysis of Time Series. North-Holland, 248 pp. I. G. Zurbenko, P. S. Porter, Construction of high-resolution wavelets, Signal Processing 65: 315-327, 1998. W. Yang, I. G. Zurbenko, A semi-adaptive smoothing algorithm in bispectrum estimation, Proceedings of the American Statistical Association, Seattle, 2006. Wei Yang and Igor Zurbenko, kzft: Kolmogorov-Zurbenko Fourier Transform and Applications, R-Project 2007.

See Also

`kzft`, `kzp`,

Examples

```
t<-1:10000
y<-2*sin(2*pi*0.1*t)+3*sin(2*pi*0.2*t) + 10*rnorm(length(t))

a<-kztp(y,50,1)
z<-log(Mod(a))
#z<-smooth.kzp(z)

omega<-seq(0,1,length=51)[2:26]
#filled.contour(omega,omega,s,xlab="freq",ylab="freq",main="Smoothed 3rd Order Periodogram")
```

---

kz_decompose — *Kolmogorov-Zurbenko decompose*

Description

Kolmogorov-Zurbenko decomposition of univariate time series into long term and seasonal components.

Usage

```
kz_decompose(y, k=3, period=NULL)
## S3 method for class 'kzd'
fitted(object, ...)
## S3 method for class 'kzd'
residuals(object, ...)
## S3 method for class 'kzd'
trend(object, ...)
## S3 method for class 'kzd'
frequency(x, ...)
## S3 method for class 'kzd'
predict(object, n.ahead = 1, prediction.interval = FALSE, level = 0.95, ...)
## S3 method for class 'kzd'
plot(x, ...)
## S3 method for class 'kzd'
summary(object, digits=getOption("digits"), ...)
```

Arguments

`y`	univariate time series.
`k`	Number of iterations.
`period`	Vector with the window sizes for the long term and seasonal period.
`object`	kzd object returned from kz_decompose.
`x`	kzd object returned from kz_decompose.
`...`	Other parameters.
`n.ahead`	The number of data points ahead to predict.

`prediction.interval`
: prediction interval level

`level`
: level

`digits`
: precision

Details

KZ is an iterated moving average filter.

Examples

```
yrs <- 20
t <- seq(0,yrs,length=yrs*365)
amp<-10
y <- amp*sin(2*pi*t*12+10)
y<-ts(y,start=c(1980,10),frequency=365)
signal<-y

tr<-seq(1, amp/2, length.out=7300)
y.orig<-signal+exp(tr)+300
y<-y.orig
set.seed(6); e <- rnorm(n = length(t), sd = 50)
SNR<-mean(abs(y)^2)/(mean((abs(e)^2)))
y<-log(y+e)

a<-kz_decompose(y, period=c(365,30))
plot(a)
```

`periodogram` *Periodogram*

Description

Raw periodogram.

Usage

```
periodogram(y)
```

Arguments

`y`
: The raw data.

Details

Periodogram is an estimate of the spectral density using FFT.

See Also

kzp,

Examples

```
t<-1:1000
f1<-0.3
f2<-0.4
noise<-15*rnorm(length(t))
s<-3*sin(2*pi*f1*t)+3*sin(2*pi*f2*t)
plot(periodogram(s+noise),type='l')
```

Index