Calculating Coefficients of the one-step predictors with the Innovations Algorithm IA()

Daniela Stängle

Introduction

The Innovations Algorithm is a recursive algorithm that can be applied to stationary time series $\{X_t\}$ with zero-mean and finite second moments. The algorithm gives the coefficients of the one-step predictors $\hat{X}_1, \hat{X}_2 \ldots$, which can be calculated recursively once the coefficients have been determined. Additionally the algorithm calculates the mean squared errors ν_i.

Example 1: Basic Usage

To apply the algorithm to a time series X, we call the function `IA()` with parameter X

```
X <- stats::rnorm(5, mean = 0, sd = 1)
out <- zeitreihen::IA(X)
```

We obtain as output the vector `nu`, which contains the mean squared errors,

```
out$nu
```

```
## [1] 1.5489753 1.5139482 1.0236729 0.9916216 0.8621950
```

the matrix $\Theta_n = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ \theta_{11} & 0 & 0 & 0 & 0 \\ \theta_{22} & \theta_{21} & 0 & 0 & 0 \\ \theta_{33} & \theta_{32} & \theta_{31} & 0 & 0 \\ \theta_{44} & \theta_{43} & \theta_{42} & \theta_{41} & 0 \end{pmatrix}$,

```
out$theta
```

```
##             [,1]       [,2]       [,3]       [,4] [,5]
## [1,]  0.00000000  0.0000000  0.0000000  0.0000000    0
## [2,] -0.15037643  0.0000000  0.0000000  0.0000000    0
## [3,] -0.53358699 -0.2359509  0.0000000  0.0000000    0
## [4,]  0.09453594 -0.5313873 -0.3366456  0.0000000    0
## [5,]  0.08942748  0.1104821 -0.6966424 -0.4005722    0
```

and the vector of coefficients, `coeffs` = $\begin{pmatrix} \theta_{41} & \theta_{42} & \theta_{43} & \theta_{44} \end{pmatrix}$,

```
out$coeffs
```

```
## [1] -0.40057216 -0.69664243  0.11048207  0.08942748
```

which can be used to calculate the one-step predictor $\hat{X}_{n+1}$ via

$$\hat{X}_{n+1} = \begin{cases} 0, & \text{if } n = 0, \\ \sum_{j=1}^{n} \theta_{nj} \left(X_{n+1-j} - \hat{X}_{n+1-j} \right), & \text{if } n = 1, 2, \ldots, \end{cases}.$$

For example the coefficient θ_{41} is the coefficient before the difference of the value of the time series $\{X_t\}$ at time $t = 4$ and its predictor $\hat{X}_4$.

Example 2: Using the optional parameter `max_lag`

Additionally, the parameter `max_lag` (per default the length of the time series) can be set, to indicate till where the θ 's should be calculated.

```
X <- stats::rnorm(100, mean = 0, sd = 1)
out <- zeitreihen::IA(X, max_lag = 6)
out$nu
```

```
## [1] 1.071010 1.062154 1.049066 1.047838 1.038345 1.014679
```

```
out$theta
```

```
##             [,1]        [,2]        [,3]        [,4]        [,5] [,6]
## [1,]  0.00000000  0.00000000  0.00000000  0.00000000  0.00000000    0
## [2,] -0.09093659  0.00000000  0.00000000  0.00000000  0.00000000    0
## [3,]  0.11835617 -0.08084221  0.00000000  0.00000000  0.00000000    0
## [4,] -0.05316949  0.11446771 -0.07704498  0.00000000  0.00000000    0
## [5,] -0.07310787 -0.06031645  0.12472879 -0.08030106  0.00000000    0
## [6,]  0.14967025 -0.05999348 -0.07727716  0.13010775 -0.06593115    0
```

Example 3: Using incorrect values

There are several inputs which are not allowed and give the following warnings:
1. If the time series contains values which are `NaN` or `Inf`:

```
X <- NaN + stats::rnorm(10, mean = 0, sd = 1)
Y <- c(0,0,0,Inf)
out <- zeitreihen::IA(Y)
```

```
## Error in zeitreihen::IA(Y): X must be filled with finite values.
```

```
out <- zeitreihen::IA(X)
```

```
## Error in zeitreihen::IA(X): X must be filled with finite values.
```

2. If the time series contains values which are not numeric or complex values:

```
X <- "2.01"
out <- zeitreihen::IA(X)
```

```
## Error in zeitreihen::IA(X): X must be only contain numeric or complex values
```

3. If the time series is not an atomic vector:

```
X <- list(m=matrix(1,nrow = 10,ncol = 5), "test")
out <- IA(X)
```

```
## Error in IA(X): X must be a native vector
```

4. If the `max_lag` is smaller than 3, not an integer or bigger than the length of the time series:

```
X <- stats::rnorm(10,0,1)
out <- zeitreihen::IA(X,2)
```

```
## Error in zeitreihen::IA(X, 2): max_lag cannot be smaller than 3
```

```r
zeitreihen::IA(X,3.14)
```

```
## Error in zeitreihen::IA(X, 3.14): max_lag must be an integer.
```

```r
zeitreihen::IA(X,51)
```

```
## Error in zeitreihen::IA(X, 51): max_lag cannot exceed length(X)
```