IA

Daniela Stängle

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

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)
```

```
## Warning in zeitreihen::IA(X): Please note: This algorithm works for stationary time series with zero
## For any other time series, the results may be incorrect.
```

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

```
out$nu
```

```
## [1] 0.5419426 0.5325773 0.3857367 0.3629487 0.3351741
```

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,]  0.000000000  0.00000000  0.0000000  0.0000000    0
## [2,] -0.131456823  0.00000000  0.0000000  0.0000000    0
## [3,] -0.498732638 -0.20048316  0.0000000  0.0000000    0
## [4,] -0.003986148 -0.50803596 -0.3281094  0.0000000    0
## [5,]  0.134175609  0.01389222 -0.6028349 -0.3953465    0
```

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

```
out$coeffs
```

```
## [1] -0.39534653 -0.60283485  0.01389222  0.13417561
```

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] 0.8781382 0.8653281 0.8547284 0.8536035 0.8357280 0.8353515
```

```
out$theta
```

```
##              [,1]        [,2]        [,3]       [,4]      [,5] [,6]
## [1,]  0.000000000  0.00000000  0.00000000  0.0000000 0.0000000    0
## [2,]  0.120780232  0.00000000  0.00000000  0.0000000 0.0000000    0
## [3,] -0.094474275  0.13414779  0.00000000  0.0000000 0.0000000    0
## [4,] -0.061351656 -0.08835309  0.13013269  0.0000000 0.0000000    0
## [5,] -0.143036899 -0.04472811 -0.10487063  0.1248829 0.0000000    0
## [6,] -0.006061999 -0.14441138 -0.04400767 -0.1047724 0.1279544    0
```

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

```
## Warning in IA(X): Please note: This algorithm works for stationary time series with zero-mean.
## For any other time series, the results may be incorrect.
```

```
## $coeffs
## [1] -0.83731516 -0.23262144  0.28811286 -0.08168933
##
## $nu
## [1] 1.2868902 1.0574253 0.7464692 0.7079188 0.6538137
##
## $theta
##             [,1]       [,2]       [,3]       [,4] [,5]
## [1,]  0.00000000  0.0000000  0.0000000  0.0000000    0
## [2,] -0.42226727  0.0000000  0.0000000  0.0000000    0
## [3,] -0.26727768 -0.6512549  0.0000000  0.0000000    0
## [4,]  0.27123429 -0.1858904 -0.7744897  0.0000000    0
## [5,] -0.08168933  0.2881129 -0.2326214 -0.8373152    0
```

Example 3: Using incorrect values

There are several inputs which are not allowed. Here are a few examples of the error messages:
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 may not contain Inf or -Inf values
```

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

```
## Error in zeitreihen::IA(X): X may not contain NAs
```

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

```
X <- c("2.01", 1, 0)
out <- zeitreihen::IA(X)
```

```
## Error in zeitreihen::IA(X): X must 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 an atomic 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)
```

```
## Warning in zeitreihen::IA(X, 2): Please note: This algorithm works for stationary time series with z
## For any other time series, the results may be incorrect.
```

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

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

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

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