The Innovations Algorithm

Introduction

STAT 464 / 864 | Fall 2024

Discrete Time Series Analysis

Skyepaphora Griffith, Queen's University

Innovations | RVs representing noise in a time series model

- The Y_t in our classical decomp.
- The Z_t in our AR(1) models
- Prediction residuals, in the innovations algorithm's context
- The j^{th} innovation is $(X_j - \hat{X}_j)$

Innovations Algorithm recursively computes:

$\hat{X}_{N+1}$

+ corresponding MMSEs $\{\nu_i\}_{i=0}^{N}$

MMSE of $\hat{X}_{N+1}$:

Innovations | RVs representing noise in a time series model

Algorithm expresses $\hat{X}_{N+1}$ as a linear combo of innovations **1** to **N**

Note:
we can write $\hat{X}_{N+1}$ in this form because $\hat{X}_j$, for **j ≤ N,** is some linear combo of $X_N, \dots X_1$

Innovations Algorithm
recursively computes:

$$P_0X_1, P_1X_2, \ldots, \boxed{P_N X_{N+1}}$$

$\hat{X}_{N+1}$

+ corresponding MMSEs $\{\nu_i\}_{i=0}^{N}$

$$\nu_0, \hat{X}_1, \nu_1, \hat{X}_1, \nu_1, \ldots, \nu_N, \hat{X}_{N+1}$$

MMSE of $\hat{X}_{N+1}$:

$$\nu_N \stackrel{\text{def}}{=} \mathrm{E}[(X_{N+1} - \hat{X}_{N+1})^2]$$

THE ALGORITHM

$$\hat{X}_{N+1} = \theta_{N,1}(X_N - \hat{X}_N) + \cdots + \theta_{N,N}(X_1 - \hat{X}_1)$$

Step 0: Initialize ν_0

$\downarrow$

THE ALGORITHM

Step 0: Initialize $\nu_0 =$

Step i:

```
For i in (1,...,N):
```

1)

$$\begin{bmatrix} \theta_{1,1} & & & & \\ \theta_{2,1} & \theta_{2,2} & & & \\ \theta_{3,1} & \theta_{3,2} & \theta_{3,3} & & \\ \vdots & \ddots & \ddots & \ddots & \\ \theta_{N,1} & \cdots & & \theta_{N,N\text{-}1} & \theta_{N,N} \end{bmatrix} \quad \begin{bmatrix} \nu_1 \\ \nu_2 \\ \nu_3 \\ \vdots \\ \nu_N \end{bmatrix}$$

2)

Psst... What does this look like?

Remark 1 | Why do we initialize with Var(X_1) ?

$$\nu_0 = \mathrm{Var}(X_1) = \gamma_X(0)$$

In step 0, the best LP of X_1 based on no predictors – i.e.) $P_0 X_1$

is the constant **a** which minimizes

$$\mathrm{E}[(X_1 - a)^2]$$

Differentiate w.r.t. **a**, set equal to 0

Remark 2 | Wait, we don't input the actual series, $\{X_t\}$?

Only input to the algorithm: ACVF at lags **{h = 0, ..., N-1}**

Sample ACVF is unreliable at lags near N

- 🕑 Not good as input for this algorithm :(
- 🕑 Limited to prediction for time series where a **parametric model** is assumed

↓

ACVF either **assumed known** at all lags,
or reliably estimated through **estimation of the model parameters**

Remark 3 | So the algorithm only works for stationary series?

We've been working in a **stationary** framework, this whole time.

There is a more general version, applicable to non-stationary series...

- 🕑 Required input: covariance function (2 variables: t_1 and t_2)
- 🕑 The text gives the version of this algorithm in section 2.5.4

This version is limited, however:

It's often infeasible to estimate $\gamma_X(t_1, t_2)$ without strict assumptions.