From Theory to Application: Developing MSARM, an R Package for Markov-Switching Autoregressive Models

Bachelor Thesis

Department of Economics
Chair of Statistics
University of Mannheim

Abstract

This thesis addresses two main objectives. First, it presents a systematic overview of the theory behind Markov-Switching models, focusing on applying the Expectation-Maximization (EM) algorithm to a broad class of Markov-Switching Autoregressive (AR) processes. Second, the thesis introduces MSARM[1], a new R package for estimating such models. Results from approximately 300 simulations show that MSARM's estimation algorithm is more robust than MSwM's, especially in more generalized settings where MSwM often fails.

Submitted to:
Dr. Ingo Steinke

Submitted by:
Julian Herbert Müller

Degree Programme: Bachelor of Science in Economics (B.Sc.)

Mannheim, 07.07.2025

[1] MSARM can be installed with the command: devtools::install_github("jmuelleo/MSARM").

Contents

1 Introduction **1**

2 Brief Remarks on Notation **1**

3 Markov-Chains and Autoregressive Processes **1**
- 3.1 Markov-Chains . . . 1
- 3.2 Autoregressive (AR) Processes . . . 4

4 Markov-Switching Models (MSM) **5**
- 4.1 Introduction to Markov-Switching Models . . . 5
- 4.2 Optimal Inference of the Regimes and Derivation of the Log-Likelihood . . . 6
- 4.3 Smoothed Inference over the Regimes . . . 6
- 4.4 Optimisation of the Conditional Log-Likelihood . . . 7
  - 4.4.1 General EM Algorithm Theory . . . 7
  - 4.4.2 Application of the EM Algorithm to Markov-Switching AR Models . . . 10
  - 4.4.3 Example 0: Switching Coefficients and Intercept, Non-Switching σ^2 . . . 11
  - 4.4.4 Example 1: Non-Switching Intercept . . . 14
  - 4.4.5 Example 2: Switching Intercept and Non-Switching Coefficients . . . 16
  - 4.4.6 Example 3: All Parameters Switch . . . 18
  - 4.4.7 Example 4: Arbitrary Subset-Switching of (c, ϕ) and Non-Switching σ^2 . . . 19
  - 4.4.8 Example 5: Arbitrary Subset-Switching of (c, ϕ) and Switching σ^2 . . . 21
- 4.5 Forecasting with Markov-Switching Models . . . 24
- 4.6 Regime Forecasting with Markov-Switching Models . . . 24

5 MSwM - The current R standard **25**

6 Building MSARM - Implementation Considerations **25**
- 6.1 MSARM.fit . . . 26
- 6.2 MSARM.predict . . . 28
- 6.3 MSARM.plot . . . 28

7 MSARM vs. MSwM **29**
- 7.1 Example 0: Switching Coefficients and Intercept, Non-Switching σ^2 . . . 29
- 7.2 Example 1: Non-Switching Intercept . . . 31
- 7.3 Example 2: Switching Intercept and Non-Switching Coefficients . . . 32
- 7.4 Example 3: All Parameters Switch . . . 33
- 7.5 Example 4: Arbitrary Subset Switching of (c, ϕ) and Non-Switching σ^2 . . . 35
- 7.6 Example 5: Arbitrary Subset-Switching of (c, ϕ) and Switching σ^2 . . . 36

8 Conclusion **37**

9 Appendix **38**
- 9.1 Optimal Inference of the Regimes and Derivation of the Log-Likelihood . . . 38
- 9.2 Smoothed Inference over the Regimes . . . 39
- 9.3 EM Algorithm for Autoregressive Processes with finite lag order . . . 42
- 9.4 Stress Testing MSARM and MSwM: Results of 288 Random Processes . . . 48

List of Figures

1 MSARM.fit: Regime Probability Plots . . . 27
2 MSARM.fit: In-Sample Fit . . . 27
3 MSARM.plot . . . 29
4 Example 0: Simulation . . . 30
5 Example 0: Regime Probability MSARM vs MSwM . . . 30
6 Example 1: Simulation . . . 31
7 Example 1: Regime Probability MSARM vs MSwM . . . 31
8 Example 2: Simulation . . . 32
9 Example 2: Regime Probability MSARM vs MSwM . . . 33
10 Example 3: Simulation . . . 34
11 Example 3: Regime Probability MSARM vs MSwM . . . 34
12 Example 4: Simulation . . . 35
13 Example 4: Regime Probability MSARM vs MSwM . . . 35
14 Example 5: Simulation . . . 36
15 Example 5: Regime Probability MSARM vs MSwM . . . 37
16 Example 0: 48 Random Processes . . . 49
17 Example 1: 48 Random Processes . . . 50
18 Example 2: 48 Random Processes . . . 51
19 Example 3: 48 Random Processes . . . 51
20 Example 4: 48 Random Processes . . . 52
21 Example 5: 48 Random Processes . . . 53

List of Tables

1 Example 0: Parameter Values . . . 29
2 Example 0: Estimated Parameter Values . . . 30
3 Example 0: Performance Metrics . . . 30
4 Example 1: Parameter Values . . . 31
5 Example 1: Estimated Parameter Values . . . 32
6 Example 1: Performance Metrics . . . 32
7 Example 2: Parameter Values . . . 32
8 Example 2: Estimated Parameter Values . . . 33
9 Example 2: Performance Metrics . . . 33
10 Example 3: Parameter Values . . . 34
11 Example 3: Esimated Parameter Values . . . 34
12 Example 3: Performance Metrics . . . 35
13 Example 4: Parameter Values . . . 35
14 Example 4: Estimated Parameter Values . . . 36
15 Example 4: Performance Metrics . . . 36
16 Example 5: Parameter Values . . . 36
17 Example 5: Estimated Parameter Values . . . 37
18 Example 5: Performance Metrics . . . 37
19 Example 0: Package Failure and Ljung-Box Test Results . . . 49
20 Example 1: Package Failure and Ljung-Box Test Results . . . 50
21 Example 2: Package Failure and Ljung-Box Test Results . . . 51
22 Example 3: Package Failure and Ljung-Box Test Results . . . 52
23 Example 4: Estimation Results and Ljung-Box Test Outcomes . . . 52
24 Example 5: Estimation Results and Ljung-Box Test Outcomes . . . 53

1 Introduction

This thesis introduces both the theoretical foundations of Markov-Switching models and their practical implementation in the newly developed R package MSARM. A systematic presentation of the underlying theory is especially valuable given that Hamilton, who pioneered these models in his seminal 1989 paper "A New Approach to the Economic Analysis of Nonstationary Time Series and the Business Cycle", presents derivations in a rather fragmented manner across several works, see Hamilton (1989, 1990, 1994), often with limited algebraic detail. The first part of this thesis therefore offers a concise and accessible summary of Hamilton's framework, tailored for undergraduate readers. Building on this, the second part presents MSARM, which implements and generalizes Hamilton's application of the Expectation-Maximization (EM) algorithm to cover a broader class of AR model structures. To the best of the author's knowledge, this systematic extension and its practical implementation within the R ecosystem are novel contributions. Compared to MSwM, a widely used alternative package that relies on different estimation techniques and often fails in more generalized scenarios, MSARM offers greater robustness, especially in those more generalized cases. This is demonstrated through approximately 300 simulations comparing both packages.

2 Brief Remarks on Notation

We start with a short discussion of the notation utilized in this paper. The density function of a continuous random variable Y will be denoted as $f_Y(y)$, the conditional density function, based on another random variable X will be denoted as $f_{Y|X}(y|x)$. We want to emphasize that in this notation, the subscript denotes the arguments of the density function, and the values in the parentheses denote the point at which the density function is evaluated. Furthermore we define a generalized density function for a vector of a discrete random variable S and a continuous random variable Y as $f_{Y,S}(y,s) = f_{Y|S}(y|s) \cdot P(S = s)$. Last but not least it should be noted that we will denote a parameter vector θ that influences the distribution of a random variable with a semicolon after the arguments in the density function's subscript. For probability functions, the parameter vector will also be denoted with a subscript. For example, for a continuous random variable Y, we could write $f_{Y;\theta}(y)$ or for a discrete random variable S, $P_\theta(S = s)$. It should be noted that both could be seen as functions in θ.

3 Markov-Chains and Autoregressive Processes

3.1 Markov-Chains

With the general notation out of the way, we can start to build the foundation of Markov-Switching AR models. For that, we start with Markov-Chains in general. Markov-Chains are stochastic processes satisfying the so-called Markov property. More precisely, we consider a sequence of random variables $S_t, S_{t-1}, ...$ that can take values in the set $\{1, \dots, N\}$, and fulfill the property:

$$P(S_t = j \mid S_{t-1} = i, \dots, S_1 = a_1) = P(S_t = j \mid S_{t-1} = i). \tag{1}$$

This means that the next state is conditionally independent of all previous states given the current state. The transition probabilities between the states of such a Markov-Chain are usually summarized in a so-called transition matrix Π, where the ith row and jth column element of Π is given by

$$(\Pi)_{i,j} = \pi_{i,j} = P(S_t = j \mid S_{t-1} = i).$$

Thereby $(\cdot)_{i,j}$ indicates the ith row and jth column element. Additionally, we want to point out that we will denote the transpose of a matrix A as A' throughout this thesis. Therefore, Π is defined such that the rows indicate the previous state and the columns represent the state being transitioned into. Furthermore, the sum of the row elements must equal 1. It has to be noted that we can represent a Markov-Chain as a Vector-Autoregressive-Process (VAR). The following discussion of representing Markov-Chains as a VAR closely follows Hamilton (1994, page 678-680). Suppose we have an underlying Markov-Chain $S_t, S_{t-1}, ...$, and the state space is $\{1, \ldots, N\}$. We then define:

$$\zeta_t = \begin{cases} (1,0,...,0)', & \text{if } S_t = 1 \\ (0,1,...,0)', & \text{if } S_t = 2 \\ ... & \\ (0,0,...,1)', & \text{if } S_t = N \end{cases}. \tag{2}$$

Thus for $S_t = i$, ζ_t equals the ith column of I_N. If $S_t = i$, then the jth element of ζ_{t+1} is a random variable with $P((\zeta_{t+1})_j = 1 | S_t = i) = \pi_{i,j}$. Thus

$$E(\zeta_{t+1}|S_t = i) = \begin{pmatrix} \pi_{i,1} \\ \pi_{i,2} \\ ... \\ \pi_{i,N} \end{pmatrix}. \tag{3}$$

Furthermore, one should note that $E(\zeta_{t+1}|S_t = i)$ is the ith column of Π'. Knowing for $S_t = i$, ζ_t is equal to the ith column of I_N it follows that $E(\zeta_{t+1}|\zeta_t) = \Pi'\zeta_t$. Due to (1) it holds that $E(\zeta_{t+1}|\zeta_t, \zeta_{t-1}, ...) = E(\zeta_{t+1}|\zeta_t) = \Pi'\zeta_t$, therefore we can write:

$$\zeta_{t+1} = \Pi'\zeta_t + v_{t+1}; \quad \text{where} \quad v_{t+1} = \zeta_{t+1} - E(\zeta_{t+1}|\zeta_t, \zeta_{t-1}, ...), \tag{4}$$

(4) implicates that:

$$\zeta_{t+m} = (\Pi')^m \zeta_t + (\Pi')^{m-1} v_{t+1} + ... + \Pi' v_{t+m-1} + v_{t+m}. \tag{5}$$

This is due to the following derivation:

$$\begin{aligned} \zeta_{t+m} &= \Pi'\zeta_{t+m-1} + v_{t+m} \\ &= \Pi'(\Pi'\zeta_{t+m-2} + v_{t+m-1}) + v_{t+m} \\ &= \Pi'(\Pi'(\Pi'\zeta_{t+m-3} + v_{t+m-2}) + v_{t+m-1}) + v_{t+m} \\ &= ... \\ &= (\Pi')^m \zeta_t + (\Pi')^{m-1} v_{t+1} + ... + \Pi' v_{t+m-1} + v_{t+m}. \end{aligned}$$

An m-period ahead forecast for a Markov-Chain can therefore be constructed in the following way:

$$E(\zeta_{t+m}|\zeta_t,\zeta_{t-1},...) = (\Pi')^m \zeta_t. \tag{6}$$

This holds because:

$$\begin{aligned}
E(\zeta_{t+m}|\zeta_t,\zeta_{t-1},...) &= E(\Pi'\zeta_{t+m-1} + v_{t+m}|\zeta_t,\zeta_{t-1},...) \\
&= E((\Pi')^m\zeta_t + (\Pi')^{m-1}v_{t+1} + ... + \Pi' v_{t+m-1} + v_{t+m}|\zeta_t,\zeta_{t-1},...) \\
&= (\Pi')^m E(\zeta_t|\zeta_t,\zeta_{t-1},...) + (\Pi')^{m-1}E(\zeta_{t+1} - E(\zeta_{t+1}|\zeta_t,\zeta_{t-1},...)|\zeta_t,\zeta_{t-1},...) + \\
&\quad ... \\
&\quad + \Pi' E(\zeta_{t+m-1} - E(\zeta_{t+m-1}|\zeta_{t+m-2},\zeta_{t-m-3},...)|\zeta_t,\zeta_{t-1},...) \\
&\quad + E(\zeta_{t+m} - E(\zeta_{t+m}|\zeta_{t+m-1},\zeta_{t+m-2},...)|\zeta_t,\zeta_{t-1},...) \\
&= (\Pi')^m\zeta_t + (\Pi')^{m-1}[E(\zeta_{t+1}|\zeta_t,\zeta_{t-1},...) - E(E(\zeta_{t+1}|\zeta_t,\zeta_{t-1},...)|\zeta_t,\zeta_{t-1},...)] + \\
&\quad ... \\
&\quad + \Pi'[E(\zeta_{t+m-1}|\zeta_t,\zeta_{t-1},...) - E(E(\zeta_{t+m-1}|\zeta_{t+m-2},\zeta_{t-m-3},...)|\zeta_t,\zeta_{t-1},...)] \\
&\quad + E(\zeta_{t+m}|\zeta_t,\zeta_{t-1},...) - E(E(\zeta_{t+m}|\zeta_{t+m-1},\zeta_{t+m-2},...)|\zeta_t,\zeta_{t-1},...) \\
&= (\Pi')^m\zeta_t.
\end{aligned}$$

We could now also condition on other random variables, like $(Y_t, Y_{t-1}, ...)$. We summarize these random variables in a vector $\mathscr{Y}_t$:

$$\mathscr{Y}_t = (Y_t, Y_{t-1}, ...), \tag{7}$$

the realisation of $\mathscr{Y}_t$ will be denoted as:

$$\vec{y}_t = (y_t, y_{t-1}, ...). \tag{8}$$

Furthmore we define:

$$\mathscr{Y}_{t:\tau} = (Y_t, Y_{t-1}, ..., Y_\tau), \tag{9}$$

the realisation of $\mathscr{Y}_{t:\tau}$ will be denoted as:

$$\vec{y}_{t:\tau} = (y_t, y_{t-1}, ..., y_\tau). \tag{10}$$

Generally speaking, $\mathscr{Y}_T$ will be the total time series of interest and $\vec{y}_T$ the observed realization. If the process is governed by regime $S_t = j$ at date t then the conditional density of Y_t is assumed to be given by $f_{Y_t|S_t,\mathscr{Y}_{t-1};\alpha}(y_t|j,\vec{y}_{t-1})$. Thereby α is a vector of parameters characterizing the conditional density function. Furthermore it is assumed that the conditional density depends only on the current regime S_t and not on past regimes, to be more precise it shall hold:

$$f_{Y_t|S_t,\mathscr{Y}_{t-1};\alpha}(y_t|s_t,\vec{y}_{t-1}) = f_{Y_t|S_t,S_{t-1},...,\mathscr{Y}_{t-1};\alpha}(y_t|s_t,s_{t-1},...,\vec{y}_{t-1}). \tag{11}$$

Additionally, S_{t+m} shall be conditonally independent of $\mathscr{Y}_t$ given S_t, for $m \geq 1$, therefore it shall hold that:

$$P_\theta(S_{t+m} = j|S_t = i) = P_\theta(S_{t+m} = j|S_t = i, \mathscr{Y}_t = \vec{y}_t). \tag{12}$$

One could now condition on this random vector $\mathscr{Y}_t$, given a parameter vector θ, which includes the transition probabilities of the Markov-Chain, as well as the parameters of the distribution of Y_t, therefore $\theta = (\Pi, \alpha)$. It should be noted that Π has to be understood in this notation, as part of θ, as the vector of transition probabilities, instead of the matrix of transition probabilities. Still, we believe that this notation improves the readability and understanding of what θ is compared to other notations. We can now write:

$$\hat{\xi}_{t|t} = E_\theta(\zeta_t|\mathscr{Y}_t = \vec{y}_t) = \begin{pmatrix} P_\theta(S_t = 1|\mathscr{Y}_t = \vec{y}_t) \\ P_\theta(S_t = 2|\mathscr{Y}_t = \vec{y}_t) \\ \dots \\ P_\theta(S_t = N|\mathscr{Y}_t = \vec{y}_t) \end{pmatrix}. \tag{13}$$

We now want to estimate ζ_{t+m} with $\hat{\xi}_{t+m|t} = E_\theta(\zeta_{t+m}|\mathscr{Y}_t = \vec{y}_t)$. From earlier we know that:

$$\begin{aligned}
(E_\theta(\zeta_{t+m}|\mathscr{Y}_t = \vec{y}_t))_j &= P_\theta(S_{t+m} = j|\mathscr{Y}_t = \vec{y}_t) \\
&= \sum_{i=1}^{N} P_\theta(S_{t+m} = j, S_t = i|\mathscr{Y}_t = \vec{y}_t) \\
&= \sum_{i=1}^{N} P_\theta(S_{t+m} = j|S_t = i, \mathscr{Y}_t = \vec{y}_t) P_\theta(S_t = i|\mathscr{Y}_t = \vec{y}_t) \\
&= \sum_{i=1}^{N} P_\theta(S_{t+m} = j|S_t = i) P_\theta(S_t = i|\mathscr{Y}_t = \vec{y}_t) \\
&= ((\Pi')^m)_{j,} \hat{\xi}_{t|t}.
\end{aligned}$$

Applying this to all elements we end up with:

$$E_\theta(\zeta_{t+m}|\mathscr{Y}_t = \vec{y}_t) = (\Pi')^m \hat{\xi}_{t|t}, \tag{14}$$

compare Hamilton (1994, page 693). This property will be essential later, therefore it is important to keep in mind that this indeed holds true.

3.2 Autoregressive (AR) Processes

Working with time series can be tricky, as one always deals with stochastic processes. The idea is the following, if one observes a time series $\vec{y}_T = (y_1, ..., y_T)$, then the observations are the realizations of the random variables $\mathscr{Y}_T = (Y_1, ..., Y_T)$. These random variables, are connected to each other, since they all stem from the same underlying stochastic process. The goal is now to estimate said stochastic process. Before estimating the parameters of a process, it is necessary to decide which process to assume as the underlying (or at least sufficiently similar) process. A rather often utilized process-family are AR processes, an AR(m) has the form:

$$Y_t = c + \phi_1 Y_{t-1} + ... + \phi_m Y_{t-m} + U_t; \quad \text{where} \quad U_t \sim WN(0, \sigma^2).$$

Thereby $WN(0, \sigma^2)$ denotes a zero-mean white noise process with variance σ^2. The white noise distribution that is most often used is the normal distribution. Therefore

$$Y_t = c + \phi_1 Y_{t-1} + ... + \phi_m Y_{t-m} + U_t; \quad \text{where} \quad U_t \sim N(0, \sigma^2),$$

would qualify as an AR(m). This framework of an AR(m) with gaussian white noise will be the foundation on which Markov-Switching AR models are built in the next section.

4 Markov-Switching Models (MSM)

4.1 Introduction to Markov-Switching Models

The basic idea of Markov-Switching models is that the stochastic process $Y_1, ..., Y_T$ is itself influenced by another, underlying stochastic process, in this specific case by an underlying Markov-Chain. Therefore, a Markov-Switching AR(1) could take the following form:

$$Y_t = c_{s_t} + \phi_{s_t} Y_{t-1} + U_t; \quad \text{where} \quad U_t \overset{i.i.d.}{\sim} N(0, \sigma^2). \tag{15}$$

We could alternatively write:

$$Y_t = X_t' \beta_{s_t} + U_t \quad \text{with} \quad X_t = \begin{pmatrix} 1 \\ Y_{t-1} \end{pmatrix} \quad \text{and} \quad \beta_{s_t} = \begin{pmatrix} c_{s_t} \\ \phi_{s_t} \end{pmatrix}.$$

Here S_t follows a first-order Markov-Chain and s_t is the value of the Markov-Chain at time t. Furthermore, important assumptions are that (11) and (12) hold true, that there is a maximum lag order (in this case 1) as well as that U_t shall be conditionally independent of $S_{t-1}, S_{t-2}, ...$ given S_t and that S_{t+m} shall be conditionally independent of $U_t, U_{t-1}, ...$ given S_t, for all $m \geq 1$. To put it more formally, it shall hold that:

$$f_{U_t|S_t;\alpha}(u_t|s_t) = f_{U_t|S_t,S_{t-1},...;\alpha}(u_t|s_t, s_{t-1}, ...), \tag{16}$$

$$P_\theta(S_{t+m} = s_{t+m}|S_t = s_t) = P_\theta(S_{t+m} = s_{t+m}|S_t = s_t, U_t = u_t, U_{t-1} = u_{t-1}, ...). \tag{17}$$

Additionally, a vector of the form of (7) would represent the vector of all observable variables until t. It has to be emphasized that the density of Y_t conditioned on $S_t = s_t$ and $\mathscr{Y}_{t-1} = \vec{y}_{t-1}$ has the following form:

$$f_{Y_t|S_t,\mathscr{Y}_{t-1};\alpha}(y_t|s_t, \vec{y}_{t-1}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{-(y_t - c_{s_t} - \phi_{s_t} y_{t-1})^2}{2\sigma^2} \right). \tag{18}$$

Here we can also see that (11) is indeed true for Markov-Switching AR(m) models with gaussian white noise, as long as earlier states of the Markov-Chain than the current state, s_t don't influence the parameters that describe the generation of Y_t, i.e the intercept, the coefficients and the error-term variance. For this specific AR(1) α would consist of $c_1, ..., c_N, \phi_1, ..., \phi_N$ and σ^2. We summarize the values of the conditional density functions for all potential states of the Markov-Chain in the following vector:

$$\eta_t = \begin{pmatrix} f_{Y_t|S_t,\mathscr{Y}_{t-1};\alpha}(y_t|1, \vec{y}_{t-1}) \\ ... \\ f_{Y_t|S_t,\mathscr{Y}_{t-1};\alpha}(y_t|N, \vec{y}_{t-1}) \end{pmatrix}. \tag{19}$$

Now that this model class has been introduced, we want to further investigate the question of optimal inference regarding the states of the Markov-Chain, often called regimes. In the following sections, the goal is to estimate the parameter vector $\theta = (\Pi, \alpha)$ given $\mathscr{Y}_t = \vec{y}_t$.

4.2 Optimal Inference of the Regimes and Derivation of the Log-Likelihood

But before we follow this endeavour, we peak into a world where we assume that θ is known. Given θ we want to get inference regarding the regimes of the time series. We start by summarizing the $P_\theta(S_t = j|\mathscr{Y}_t = \vec{y}_t)$ and $P_\theta(S_{t+1} = j|\mathscr{Y}_t = \vec{y}_t)$ for all $j = 1,...,N$, similarily to (13) in:

$$\hat{\xi}_{t|t} = \begin{pmatrix} P_\theta(S_t = 1|\mathscr{Y}_t = \vec{y}_t) \\ ... \\ P_\theta(S_t = N|\mathscr{Y}_t = \vec{y}_t) \end{pmatrix} \quad \text{and} \quad \hat{\xi}_{t+1|t} = \begin{pmatrix} P_\theta(S_{t+1} = 1|\mathscr{Y}_t = \vec{y}_t) \\ ... \\ P_\theta(S_{t+1} = N|\mathscr{Y}_t = \vec{y}_t) \end{pmatrix}. \tag{20}$$

The claim Hamilton makes now is that the optimal inference can be derived by iterating over the following equations:

$$\hat{\xi}_{t|t} = \frac{(\hat{\xi}_{t|t-1} \odot \eta_t)}{\mathbf{1}'(\hat{\xi}_{t|t-1} \odot \eta_t)}, \tag{21}$$

$$\hat{\xi}_{t+1|t} = \Pi' \hat{\xi}_{t|t}. \tag{22}$$

Where $\odot$ denotes element-by-element multiplication. In addition, the value of the Log-Likelihood function for the vector of all observations $\vec{y}_T$ at the point θ is gained as a side product:

$$\mathscr{L}(\theta) = \sum_{t=1}^{T} \ln(f_{Y_t|\mathscr{Y}_{t-1};\theta}(y_t|\vec{y}_{t-1})), \tag{23}$$

$$f_{Y_t|\mathscr{Y}_{t-1};\theta}(y_t|\vec{y}_{t-1}) = \mathbf{1}'(\hat{\xi}_{t|t-1} \odot \eta_t), \tag{24}$$

see Hamilton (1994, page 692). That this is indeed true is shown in the Appendix, section 9.1. Based on this system of two equations and a given θ we start with a random $\hat{\xi}_{1|0}$ and iterate over all t until we reach T (T is the number of periods for which observations of the time series exist). This gives us the regime probabilities conditionally on the data until t. Additionally we get the Log-Likelihood function, which can be optimized in θ to derive the Maximum Likelihood estimate of θ. It is important to note that a direct optimization of $\mathscr{L}(\theta)$ can be computationally expensive and often leads to suboptimal results. Therefore, Hamilton introduced, in his paper "Analysis of Time Series subject to Changes in Regime" from 1990, an iterative optimization algorithm for $\mathscr{L}(\theta)$, which is an application of the EM algorithm. Applying a specific variant of the EM algorithm to this optimization problem, instead of more general optimization algorithms can lead to better and computationally less expensive results, see Hamilton (1990, page 40-41). The details of this specific application of the EM algorithm for the optimization of $\mathscr{L}(\theta)$ are shown throughout the following sections, this includes a derivation of an algorithm developed by Kim (1994), which can improve some aspects of the application of the EM algorithm as shown in Hamilton (1990). The derivation shown of the "Kim-Algorithm" is based on Hamilton (1994, page 700-702).

4.3 Smoothed Inference over the Regimes

Before further investigating the optimization of the log-likelihood and the associated parameter estimation, we want to first discuss how to get inference on $P_\theta(S_{t-1} = i|\mathscr{Y}_T = \vec{y}_T)$, as this will be essential

for the application of the aforementioned EM algorithm. Estimating $P_\theta(S_{t-1} = i|\mathscr{Y}_T = \vec{y}_T)$ is often called "smoothed inference" for the regimes. To get this smoothed inference, we apply the algorithm from Kim (1994). The proposition is that, assuming S_t follows a first order Markov-Chain and that the conditional density, $f_{Y_t|S_t,\mathscr{Y}_{t-1};\alpha}(y_t|j,\vec{y}_{t-1})$, depends only on $S_t, S_{t-1}, ...$ through S_t, i.e. that (11) holds true, an assumption we have already made throughout section 3.1, it shall hold that:

$$\hat{\xi}_{t|T} = \hat{\xi}_{t|t} \odot (\Pi(\hat{\xi}_{t+1|T}(\div)\hat{\xi}_{t+1|t})), \tag{25}$$

where $(\div)$ is the symbol for element-by-element division, see Hamilton (1994, page 694). To get the values of $P_\theta(S_{t-1} = i|\mathscr{Y}_T = \vec{y}_T)$ for t in $1, ..., T$ one starts with $t = T-1$ and iterates backwards. The derivation can be found in the Appendix, section 9.2.

4.4 Optimisation of the Conditional Log-Likelihood

4.4.1 General EM Algorithm Theory

We now turn to the EM algorithm. Assuming we observe $\vec{y}_T = (y_1, ..., y_T)$, a trick that is often utilized is to optimize a density function of the form $f_{Y_T,...,Y_{m+1}|Y_m,...,Y_1;\lambda}(y_T, ..., y_{m+1}|y_m, ..., y_1)$ in λ instead of $f_{Y_T,...,Y_1;\theta}(y_T, ..., y_1)$ in θ. We have to optimize in λ because if we choose such a conditional likelihood function, then we have to make assumptions about how the initial states $(Y_m, ..., Y_1)$ are distributed. The simplest approach is to assume that they are seperately drawn from a distribution with the parameters ρ. Thereby ρ shall be unrelated of Π and α. The new parameter vector λ is therefore defined as $\lambda = (\Pi, \alpha, \rho)$. The conditional likelihood function is primarily chosen due to practical reasons. Optimizing the likelihood function instead is often more challenging and yields next to no additional gain. Choosing the conditional likelihood enables the application of the EM algorithm, as described by Hamilton (1990), which estimates the parameters with relatively low computational demands, at least compared to other numerical methods, see Hamilton (1990, page 40). Still it has to be emphasized that the EM algorithm that Hamilton introduces only leads to a local maximum of the conditional likelihood function, as will be shown throughout this section. Generally speaking, this is not problematic, as one can start the algorithm with several different values to see whether the results are robust. We start by defining

$$P_\lambda(S_m = s_m, S_{m-1} = s_{m-1}, ..., S_1 = s_1|Y_m = y_m, ..., Y_1 = y_1) = \rho_{s_m,...,s_1}, \tag{26}$$

and

$$\rho = (\rho_{1,1,..,1}, \rho_{1,1,..,2}, ..., \rho_{N,N,..,N}). \tag{27}$$

Thereby (26) is the vector of population probabilities, which are aggregated in (27). With this we can now derive the general EM algorithm: We assume we know nothing about λ and it should be chosen such that the conditional likelihood

$$f_{\mathscr{Y}_{T:(m+1)}|\mathscr{Y}_m;\lambda}(\vec{y}_{T:(m+1)}|\vec{y}_m) = f_{Y_T,...,Y_{m+1}|Y_m,...,Y_1;\lambda}(y_T, ..., y_{m+1}|y_m, ..., y_1), \tag{28}$$

is maximized, the optimising λ is called λ_{MLE}. Furthermore we define

$$\mathscr{S} = (S_T, S_{T-1}, ..., S_1), \tag{29}$$

$$\vec{s}_T = (s_T, s_{T-1}, ..., s_1), \tag{30}$$

$$Z_t = (S_t, S_{t-1}, ..., S_{t-m}, Y_{t-1}, ..., Y_{t-m}), \tag{31}$$

$$z_t = (s_t, s_{t-1}, ..., s_{t-m}, y_{t-1}, ..., y_{t-m}), \tag{32}$$

$$f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\lambda}(\vec{y}_{T:(m+1)}, \vec{s}_T|\vec{y}_m) = f_{Y_T,...,Y_{m+1},S_T,...,S_1|Y_m,...,Y_1;\lambda}(y_T, ..., y_{m+1}, s_T, ..., s_1|y_m, ..., y_1), \tag{33}$$

$$\sum_{\vec{s}_T} P(\mathscr{S} = \vec{s}_T) = \sum_{s_T=1}^{N} \cdots \sum_{s_1=1}^{N} P(S_T = s_T, ..., S_1 = s_1), \tag{34}$$

$$f_{\mathscr{Y}_{T:(m+1)}|\mathscr{Y}_m;\lambda}(\vec{y}_{T:(m+1)}|\vec{y}_m) = \sum_{\vec{s}_T} f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\lambda}(\vec{y}_{T:(m+1)}, \vec{s}_T|\vec{y}_m), \tag{35}$$

and

$$Q_{\lambda_l,\vec{y}_T}(\lambda_{l+1}) = \sum_{\vec{s}_T} \ln(f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\lambda_{l+1}}(\vec{y}_{T:(m+1)}, \vec{s}_T|\vec{y}_m)) f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\lambda_l}(\vec{y}_{T:(m+1)}, \vec{s}_T|\vec{y}_m). \tag{36}$$

Where we call $Q_{\lambda_l,\vec{y}_T}(\lambda_{l+1})$ the expected log-likelihood. With that remark we finish the necessary notation, which is based on Hamilton (1990, page 42-44) and Hamilton (1990, page 46-47). Now we want to show that the EM algorithm works. It is noteworthy that the EM algorithm can be seen from two different perspectives.

1. $\hat{\lambda}_l$ shall be the solution of the lth optimization problem (of a sequence of optimization problems), thereby the optimization problems are constructed in such a way that $\hat{\lambda}_{l+1}$ leads to a higher value of the conditional likelihood function than $\hat{\lambda}_l$, the limit of the $\hat{\lambda}$s leads to a local maximum of the conditional likelihood function: $\lim_{l\to\infty} \hat{\lambda}_l = \hat{\lambda}_{MLE}$, see Hamilton (1990, page 47).

2. Alternatively, one could say that if we were to observe $\mathscr{S}$ directly i.e. know $\vec{s}_T$, then the first order condition (FOC) characterizing $\hat{\lambda}_{MLE}(\vec{s}_T)$ would be:

$$\frac{\partial \ln(f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\lambda}(\vec{y}_{T:(m+1)}, \vec{s}_T|\vec{y}_m))}{\partial \lambda}\Big|_{\lambda=\hat{\lambda}_{MLE}(\vec{s}_T)} = 0.$$

   However, this set of conditions, one for each possible realization of $\mathscr{S}$, can be weighted by the probability of actually observing this particular $\vec{s}_T$. These probabilities are obtained through inference about $\mathscr{S}$ at the given step of the EM algorithm, i.e., $P_{\hat{\lambda}_l}(\mathscr{S} = \vec{s}_T|\mathscr{Y}_T = \vec{y}_T)$. The sum over all weighted conditions for a given $\hat{\lambda}_l$ characterizes the EM algorithm's update choice for $\hat{\lambda}_{l+1}$, see Hamilton (1990, page 47).

First we want to discuss the EM algorithm as a sequence of optimization problems. $\hat{\lambda}_l$ is the result of the lth optimization problem, and we start with a random $\hat{\lambda}_0$ for the first optimization problem. We choose $\hat{\lambda}_{l+1}$ such that $Q_{\hat{\lambda}_l,\vec{y}_T}(\lambda_{l+1})$ is maximized. We remember:

$$Q_{\hat{\lambda}_l,\vec{y}_T}(\lambda_{l+1}) = \sum_{\vec{s}_T} \ln(f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\lambda_{l+1}}(\vec{y}_{T:(m+1)}, \vec{s}_T|\vec{y}_m)) f_{\mathscr{Y}_{T:(m+1)}|\mathscr{Y}_m;\hat{\lambda}_l}(\vec{y}_{T:(m+1)}|\vec{y}_m).$$

Therefore $\hat{\lambda}_{l+1}$ fulfills:

$$\sum_{\vec{s}_T} \frac{\partial \ln(f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\lambda_{l+1}}(\vec{y}_{T:(m+1)},\vec{s}_T|\vec{y}_m))}{\partial \lambda_{l+1}}\Big|_{\lambda_{l+1}=\hat{\lambda}_{l+1}} \cdot f_{\mathscr{Y}_{T:(m+1)}|\mathscr{Y}_m;\hat{\lambda}_l}(\vec{y}_{T:(m+1)}|\vec{y}_m) = 0. \tag{37}$$

It holds that $\hat{\lambda}_{l+1}$ is associated with an higher value of the conditional likelihood function than $\hat{\lambda}_l$ i.e. $f_{\mathscr{Y}_{T:(m+1)}|\mathscr{Y}_m;\hat{\lambda}_{l+1}}(\vec{y}_{T:(m+1)}|\vec{y}_m) \geq f_{\mathscr{Y}_{T:(m+1)}|\mathscr{Y}_m;\hat{\lambda}_l}(\vec{y}_{T:(m+1)}|\vec{y}_m)$. In the following, we closely follow the derivation in Hamilton (1990, page 48-49). Per construction $\hat{\lambda}_{l+1}$ maximizes $Q_{\hat{\lambda}_l,\vec{y}_T}(\lambda_{l+1})$, thus:

$$Q_{\hat{\lambda}_l,\vec{y}_T}(\hat{\lambda}_{l+1}) \geq Q_{\hat{\lambda}_l,\vec{y}_T}(\hat{\lambda}_l); \quad \text{equal if} \quad \hat{\lambda}_{l+1} = \hat{\lambda}_l.$$

We also note that $\forall x \in \mathbb{R}^+ : \ln(x) \leq (x-1)$. This is because we can show that $h(x) = x - 1 - \ln(x)$ has a minimum at $x = 1$ and $h(1) = 0$. The first order condition is given by:

$$\begin{aligned} h'(x) &= 1 - \frac{1}{x} = 0 \\ &\Leftrightarrow x = 1. \end{aligned}$$

That this is indeed a minimum can be shown by checking the second derivative at the point $x = 1$:

$$h''(1) = \frac{1}{1^2} > 0,$$

thus $h(x)$ has a minimum at the point $x = 1$ and $\forall x \in \mathbb{R}^+ : \ln(x) \leq (x-1)$ is therefore a true statement. We can apply this now and show:

$$\begin{aligned} 0 &\leq Q_{\hat{\lambda}_l,\vec{y}_T}(\hat{\lambda}_{l+1}) - Q_{\hat{\lambda}_l,\vec{y}_T}(\hat{\lambda}_l) \\ &= \sum_{\vec{s}_T} \ln(f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\hat{\lambda}_{l+1}}(\vec{y}_{T:(m+1)},\vec{s}_T|\vec{y}_m)) f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\hat{\lambda}_l}(\vec{y}_{T:(m+1)},\vec{s}_T|\vec{y}_m) \\ &\quad - \sum_{\vec{s}_T} \ln(f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\hat{\lambda}_l}(\vec{y}_{T:(m+1)},\vec{s}_T|\vec{y}_m)) f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\hat{\lambda}_l}(\vec{y}_{T:(m+1)},\vec{s}_T|\vec{y}_m) \\ &= \sum_{\vec{s}_T} \ln\left[\frac{f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\hat{\lambda}_{l+1}}(\vec{y}_{T:(m+1)},\vec{s}_T|\vec{y}_m)}{f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\hat{\lambda}_l}(\vec{y}_{T:(m+1)},\vec{s}_T|\vec{y}_m)}\right] f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\hat{\lambda}_l}(\vec{y}_{T:(m+1)},\vec{s}_T|\vec{y}_m) \\ &\leq \sum_{\vec{s}_T} \left[\frac{f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\hat{\lambda}_{l+1}}(\vec{y}_{T:(m+1)},\vec{s}_T|\vec{y}_m)}{f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\hat{\lambda}_l}(\vec{y}_{T:(m+1)},\vec{s}_T|\vec{y}_m)} - 1\right] f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\hat{\lambda}_l}(\vec{y}_{T:(m+1)},\vec{s}_T|\vec{y}_m) \\ &= \sum_{\vec{s}_T} (f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\hat{\lambda}_{l+1}}(\vec{y}_{T:(m+1)},\vec{s}_T|\vec{y}_m) - f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\hat{\lambda}_l}(\vec{y}_{T:(m+1)},\vec{s}_T|\vec{y}_m)) \\ &= f_{\mathscr{Y}_{T:(m+1)}|\mathscr{Y}_m;\hat{\lambda}_{l+1}}(\vec{y}_{T:(m+1)}|\vec{y}_m) - f_{\mathscr{Y}_{T:(m+1)}|\mathscr{Y}_m;\hat{\lambda}_l}(\vec{y}_{T:(m+1)}|\vec{y}_m). \end{aligned}$$

With that we have shown that the algorithm indeed leads to an increase of the conditional likelihood function with each step. Now we want to show that if $\hat{\lambda}_{l+1} = \hat{\lambda}_l$, then the first order condition for maximizing $f_{\mathscr{Y}_{T:(m+1)}|\mathscr{Y}_m;\lambda}(\vec{y}_{T:(m+1)}|\vec{y}_m)$ is fulfilled by $\lambda = \hat{\lambda}_l$. This is the case because if:

$$\frac{\partial Q_{\hat{\lambda}_l,\vec{y}_T}(\lambda_{l+1})}{\partial \lambda_{l+1}}\Big|_{\lambda_{l+1}=\hat{\lambda}_l} = 0.$$

Then:

$$\frac{\partial f_{\mathscr{Y}_{T:(m+1)}|\mathscr{Y}_m;\lambda}(\vec{y}_{T:(m+1)}|\vec{y}_m)}{\partial \lambda}\Big|_{\lambda=\hat{\lambda}_l} = 0,$$

this holds true because:

$$\begin{aligned}
\frac{\partial Q_{\hat{\lambda}_l,\vec{y}_T}(\lambda_{l+1})}{\partial \lambda_{l+1}}\Big|_{\lambda_{l+1}=\hat{\lambda}_l} &= \sum_{\vec{s}_T}\left(\frac{1}{f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\lambda_{l+1}}(\vec{y}_{T:(m+1)},\vec{s}_T|\vec{y}_m)}\frac{\partial f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\lambda_{l+1}}(\vec{y}_{T:(m+1)},\vec{s}_T|\vec{y}_m)}{\partial \lambda_{l+1}}\right)\Big|_{\lambda_{l+1}=\hat{\lambda}_l} \\
&\quad \cdot f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\hat{\lambda}_l}(\vec{y}_{T:(m+1)},\vec{s}_T|\vec{y}_m) \\
&= \sum_{\vec{s}_T}\frac{\partial f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\lambda_{l+1}}(\vec{y}_{T:(m+1)},\vec{s}_T|\vec{y}_m)}{\partial \lambda_{l+1}}\Big|_{\lambda_{l+1}=\hat{\lambda}_l} \\
&= \frac{\partial f_{\mathscr{Y}_{T:(m+1)}|\mathscr{Y}_m;\lambda_{l+1}}(\vec{y}_{T:(m+1)}|\vec{y}_m)}{\partial \lambda_{l+1}}\Big|_{\lambda_{l+1}=\hat{\lambda}_l}.
\end{aligned}$$

With that in mind, we proceed to consider the second perspective. One could say that the EM algorithm replaces the unobserved scores with their conditional expectation. Assuming we know $\vec{s}_T$, then $\hat{\lambda}_{MLE}(\vec{s}_T)$ is characterized by the first-order condition:

$$\frac{\partial \ln(f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\lambda}(\vec{y}_{T:(m+1)},\vec{s}_T|\vec{y}_m))}{\partial \lambda}\Big|_{\lambda=\hat{\lambda}_{MLE}(\vec{s}_T)} = 0.$$

Even so we have no data regarding $\mathscr{S}$ we still have inference regarding $\mathscr{S}$, at least for the lth step, based on $\hat{\lambda}_l$ and $\mathscr{Y}_T = \vec{y}_T$. We can formulate:

$$P_{\hat{\lambda}_l}(\mathscr{S} = \vec{s}_T|\mathscr{Y}_T = \vec{y}_T) = \frac{f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\hat{\lambda}_l}(\vec{y}_{T:(m+1)},\vec{s}_T|\vec{y}_m)}{f_{\mathscr{Y}_{T:(m+1)}|\mathscr{Y}_m;\hat{\lambda}_l}(\vec{y}_{T:(m+1)}|\vec{y}_m)}.$$

For all possible values of $\mathscr{S}$, which amounts to N^T possibilities, there exists such a first-order condition. If one weights all these FOCs with the probability $P_{\hat{\lambda}_l}(\mathscr{S} = \vec{s}_T|\mathscr{Y}_T = \vec{y}_T)$ then we choose λ such that:

$$\begin{aligned}
&\sum_{\vec{s}_T}\frac{\partial \ln(f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\lambda}(\vec{y}_{T:(m+1)},\vec{s}_T|\vec{y}_m))}{\partial \lambda}\cdot\frac{f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\hat{\lambda}_l}(\vec{y}_{T:(m+1)},\vec{s}_T|\vec{y}_m)}{f_{\mathscr{Y}_{T:(m+1)}|\mathscr{Y}_m;\hat{\lambda}_l}(\vec{y}_{T:(m+1)}|\vec{y}_m)} = 0 \\
&\Leftrightarrow \frac{1}{f_{\mathscr{Y}_{T:(m+1)}|\mathscr{Y}_m;\hat{\lambda}_l}(\vec{y}_{T:(m+1)}|\vec{y}_m)}\frac{\partial Q_{\hat{\lambda}_l,\vec{y}_T}(\lambda)}{\partial \lambda} = 0 \\
&\Leftrightarrow \frac{\partial Q(\lambda,\hat{\lambda}_l,\vec{y}_T)}{\partial \lambda} = 0.
\end{aligned}$$

This again is the characterizing condition for $\hat{\lambda}_{l+1}$ in the first perspective!

4.4.2 Application of the EM Algorithm to Markov-Switching AR Models

The next essential step in the theoretical exposition is to demonstrate the application of the EM algorithm to models of the type we are using, that is, models with an underlying Markov-Chain and a dependence on a maximum lag order. Hamilton states that, when using the EM algorithm to maximize

the conditional log-likelihood, one obtains three equations to iterate over, see Hamilton (1990, page 51). The equations given by Hamilton are:

$$\pi_{i,j}^{(l+1)} = \frac{\sum_{t=m+1}^{T} P_{\lambda_l}(S_t = j, S_{t-1} = i | \mathscr{Y}_T = \vec{y}_T)}{\sum_{t=m+1}^{T} P_{\lambda_l}(S_{t-1} = i | \mathscr{Y}_T = \vec{y}_T)}, \tag{38}$$

$$\sum_{t=m+1}^{T} \sum_{s_t=1}^{N} \cdots \sum_{s_{t-m}=1}^{N} \frac{\partial \ln(f_{Y_t|Z_t;\alpha}(y_t|z_t))}{\partial \alpha}\Big|_{\alpha=\alpha_{l+1}} \cdot P_{\lambda_l}(S_t = s_t, ..., S_{t-m} = s_{t-m} | \mathscr{Y}_T = \vec{y}_T) = 0, \tag{39}$$

$$\rho_{i_m,...,i_1}^{(l+1)} = P_{\lambda_l}(S_m = i_m, ..., S_1 = i_1 | \mathscr{Y}_T = \vec{y}_T). \tag{40}$$

We show that these equations are indeed true in the Appendix, section 9.3. Given this exposition one can apply the EM algorithm to a very broad class of models, even broader than just Markov-Switching AR models, but to actually estimate a specific model, one has to specify the assumed process in more detail. Hamilton (1990) shows only one potential setup for Markov-Switching AR models, we will call this setup "Example 0" throughout this paper. Deriving the results for Example 0 will be the subject of the next section. After this is done we will show 5 more examples, establishing broader AR setups, until the theory for estimating any potential Markov-Switching AR model has been shown. Thereby, Example 1 to Example 3 are specific examples that are introduced to improve the readability of the general cases, Example 4 and Example 5.

4.4.3 Example 0: Switching Coefficients and Intercept, Non-Switching σ^2

Here we assume that the underlying process is given by a Markov-Switching AR(m), which fulfills all assumptions formulated in section 4.1, the only difference is that the assumed underlying AR process now has the following form.

$$Y_t = c_{s_t} + \phi_{1,s_t} Y_{t-1} + ... + \phi_{m,s_t} Y_{t-m} + U_t \quad \text{where} \quad U_t \overset{i.i.d.}{\sim} N(0, \sigma^2). \tag{41}$$

We could alternatively write:

$$Y_t = X_t' \beta_{s_t} + U_t \quad \text{with} \quad X_t = \begin{pmatrix} 1 \\ Y_{t-1} \\ ... \\ Y_{t-m} \end{pmatrix} \quad \text{and} \quad \beta_{s_t} = \begin{pmatrix} c_{s_t} \\ \phi_{1,s_t} \\ ... \\ \phi_{m,s_t} \end{pmatrix}.$$

The following derivation closely follows Hamilton (1990, page 56-58). First, we see that:

$$f_{Y_t|Z_t;\alpha}(y_t|z_t) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{-(y_t - x_t'\beta_{s_t})^2}{2\sigma^2} \right).$$

It is important to note that x_t denotes the vector of realizations for X_t. Then it holds that:

$$\frac{\partial \ln(f_{Y_t|Z_t;\alpha}(y_t|z_t))}{\partial \beta_j} = \begin{cases} \frac{(y_t - x_t'\beta_j)x_t}{\sigma^2}, & \text{if } S_t = j \\ 0, & \text{otherwise} \end{cases}, \tag{42}$$

$$\frac{\partial \ln(f_{Y_t|Z_t;\alpha}(y_t|z_t))}{\partial \sigma^{-2}} = \frac{\sigma^2}{2} - \frac{(y_t - x_t'\beta_{s_t})^2}{2}. \tag{43}$$

(43) is indeed true, because:

$$\begin{aligned}
\ln\left[\frac{1}{\sqrt{2\pi}\sigma}\exp\left(\frac{-(y_t - x_t'\beta_{s_t})^2}{2\sigma^2}\right)\right] &= \ln\left(\frac{1}{\sqrt{2\pi}\sigma}\right) - \frac{1}{2}\left(\frac{(y_t - x_t'\beta_{s_t})^2}{\sigma^2}\right) \\
&= \ln(1) - \ln(\sqrt{2\pi}\sigma) - \frac{1}{2}(y_t - x_t'\beta_{s_t})^2\frac{1}{\sigma^2} \\
&= -\ln(\sqrt{2\pi}) - \ln(\sigma) - \frac{1}{2}(y_t - x_t'\beta_{s_t})^2\frac{1}{\sigma^2} \\
&= -\ln(\sqrt{2\pi}) - \ln\left(\left(\frac{1}{\sigma^2}\right)^{-\frac{1}{2}}\right) - \frac{1}{2}(y_t - x_t'\beta_{s_t})^2\frac{1}{\sigma^2}.
\end{aligned}$$

And thus:

$$\begin{aligned}
\frac{\partial \ln\left[\frac{1}{\sqrt{2\pi}\sigma}\exp\left(\frac{-(y_t - x_t'\beta_{s_t})^2}{2\sigma^2}\right)\right]}{\partial\left(\frac{1}{\sigma^2}\right)} &= \left(\frac{1}{\sigma^2}\right)^{\frac{1}{2}}\left(\frac{1}{2}\left(\frac{1}{\sigma^2}\right)^{-\frac{3}{2}}\right) - \frac{1}{2}(y_t - x_t'\beta_{s_t})^2 \\
&= \frac{\sigma^2}{2} - \frac{(y_t - x_t'\beta_{s_t})^2}{2}.
\end{aligned}$$

We insert these results into (39), which leads to:

$$\sum_{t=m+1}^{T} \frac{(y_t - x_t'\beta_j^{(l+1)})x_t}{\sigma^2_{(l+1)}} P_{\lambda_l}(S_t = j|\mathscr{Y}_T = \vec{y}_T) = 0, \tag{44}$$

and

$$\sigma^2_{(l+1)} = \sum_{t=m+1}^{T}\sum_{s_t=1}^{N} \frac{(y_t\sqrt{P_{\lambda_l}(S_t = s_t|\mathscr{Y}_T = \vec{y}_T)} - x_t'\sqrt{P_{\lambda_l}(S_t = s_t|\mathscr{Y}_T = \vec{y}_T)}\beta_{s_t}^{(l+1)})^2}{(T-m)}. \tag{45}$$

(44) is true because (42) can be understood as a function of S_t, we could write

$$\frac{\partial \ln(f_{Y_t|Z_t;\alpha}(y_t|z_t))}{\partial \beta_j} = g_j(S_t) = \begin{cases} \frac{(y_t - x_t'\beta_j)x_t}{\sigma^2}, & \text{if } S_t = j \\ 0, & \text{otherwise} \end{cases},$$

thus we can write:

$$\begin{aligned}
&\sum_{t=m+1}^{T}\sum_{s_t=1}^{N}\sum_{s_{t-1}=1}^{N}\cdots\sum_{s_{t-m}=1}^{N} \frac{\partial \ln(f_{Y_t|Z_t;\alpha}(y_t|z_t))}{\partial \beta_j}\Big|_{\alpha=\alpha^{(l+1)}} P_{\lambda_l}(S_t = s_t, ..., S_{t-m} = s_{t-m}|\mathscr{Y}_T = \vec{y}_T) = 0 \\
&\Leftrightarrow \sum_{t=m+1}^{T}\sum_{s_t=1}^{N}\sum_{s_{t-1}=1}^{N}\cdots\sum_{s_{t-m}=1}^{N} g_j(s_t) P_{\lambda_l}(S_t = s_t, ..., S_{t-m} = s_{t-m}|\mathscr{Y}_T = \vec{y}_T) = 0 \\
&\Leftrightarrow \sum_{t=m+1}^{T}\sum_{s_t=1}^{N} g_j(s_t)\sum_{s_{t-1}=1}^{N}\cdots\sum_{s_{t-m}=1}^{N} P_{\lambda_l}(S_t = s_t, ..., S_{t-m} = s_{t-m}|\mathscr{Y}_T = \vec{y}_T) = 0 \\
&\Leftrightarrow \sum_{t=m+1}^{T}\sum_{s_t=1}^{N} g_j(s_t) P_{\lambda_l}(S_t = s_t|\mathscr{Y}_T = \vec{y}_T) = 0 \\
&\Leftrightarrow \sum_{t=m+1}^{T} \frac{(y_t - x_t'\beta_j^{(l+1)})x_t}{\sigma^2_{(l+1)}} P_{\lambda_l}(S_t = j|\mathscr{Y}_T = \vec{y}_T) = 0.
\end{aligned}$$

And (45) is true because:

$$\sum_{t=m+1}^{T}\sum_{s_t=1}^{N}\cdots\sum_{s_{t-m}=1}^{N}\left(\frac{\sigma^2_{(l+1)}}{2}-\frac{(y_t-x_t'\beta_{s_t}^{(l+1)})^2}{2}\right)P_{\lambda_l}(S_t=s_t,...,S_{t-m}=s_{t-m}|\mathscr{Y}_T=\vec{y}_T)=0$$

$$\Leftrightarrow \sum_{t=m+1}^{T}\sum_{s_t=1}^{N}\left(\frac{\sigma^2_{(l+1)}}{2}-\frac{(y_t-x_t'\beta_{s_t}^{(l+1)})^2}{2}\right)P_{\lambda_l}(S_t=s_t|\mathscr{Y}_T=\vec{y}_T)=0$$

$$\Leftrightarrow \sum_{t=m+1}^{T}\sum_{s_t=1}^{N}P_{\lambda_l}(S_t=s_t|\mathscr{Y}_T=\vec{y}_T)\frac{\sigma^2_{(l+1)}}{2}=\sum_{t=m+1}^{T}\sum_{s_t=1}^{N}\frac{(y_t-x_t'\beta_{s_t}^{(l+1)})^2}{2}P_{\lambda_l}(S_t=s_t|\mathscr{Y}_T=\vec{y}_T)$$

$$\Leftrightarrow (T-m)\frac{\sigma^2_{(l+1)}}{2}=\sum_{t=m+1}^{T}\sum_{s_t=1}^{N}\frac{(y_t-x_t'\beta_{s_t}^{(l+1)})^2}{2}P_{\lambda_l}(S_t=s_t|\mathscr{Y}_T=\vec{y}_T)$$

$$\Leftrightarrow \sigma^2_{(l+1)}=\sum_{t=m+1}^{T}\sum_{s_t=1}^{N}\frac{(y_t-x_t'\beta_{s_t}^{(l+1)})^2}{(T-m)}P_{\lambda_l}(S_t=s_t|\mathscr{Y}_T=\vec{y}_T)$$

$$\Leftrightarrow \sigma^2_{(l+1)}=\sum_{t=m+1}^{T}\sum_{s_t=1}^{N}\frac{(y_t\sqrt{P_{\lambda_l}(S_t=s_t|\mathscr{Y}_T=\vec{y}_T)}-x_t'\sqrt{P_{\lambda_l}(S_t=s_t|\mathscr{Y}_T=\vec{y}_T)}\beta_{s_t}^{(l+1)})^2}{(T-m)}.$$

Conveniently we can estimate in this specific case all parameters via an OLS Regression. Let us assume we have the parameter vector from the previous iteration λ_l (to start the algorithm one starts with a random λ_0), we first define:

$$y_t^*=y_t\sqrt{P_{\lambda_l}(S_t=j|\mathscr{Y}_T=\vec{y}_T)}\quad\text{and}\quad x_t^*=x_t\sqrt{P_{\lambda_l}(S_t=j|\mathscr{Y}_T=\vec{y}_T)}.$$

Then we regress y_t^* on x_t^*. Thus $\sum_{t=m+1}^{T}(y_t^*-x_t'^*\beta_j)^2$ should be minimized, which leads to the following FOC that characterises $\beta_j^{(l+1)}$:

$$\sum_{t=m+1}^{T}2(y_t^*-(x_t^*)'\beta_j^{(l+1)})(-1)x_t^*=0$$

$$\Leftrightarrow \sum_{t=m+1}^{T}(y_t\sqrt{P_{\lambda_l}(S_t=j|\mathscr{Y}_T=\vec{y}_T)}-x_t'\sqrt{P_{\lambda_l}(S_t=j|\mathscr{Y}_T=\vec{y}_T)}\beta_j^{(l+1)})x_t\sqrt{P_{\lambda_l}(S_t=j|\mathscr{Y}_T=\vec{y}_T)}=0$$

$$\Leftrightarrow \sum_{t=m+1}^{T}(y_t-x_t'\beta_j^{(l+1)})x_tP_{\lambda_l}(S_t=j|\mathscr{Y}_T=\vec{y}_T)=0.$$

Which is equivalent to conditon (44). This is now done N times to estimate $\beta_1^{(l+1)},...,\beta_N^{(l+1)}$. By squaring and summing the residuals we get our estimate of σ^2:

$$\sigma^2_{(l+1)}=\frac{1}{(T-m)}\sum_{j=1}^{N}\sum_{t=m+1}^{T}(y_t^*-(x_t^*)'\beta_j^{(l+1)})^2.$$

Summarizing one can say, that we achieve our estimate for the β_j of the next iteration by solving the following optimization problem:

$$\arg\min_{\beta_j}\sum_{t=m+1}^{T}(y_t^*-x_t'^*\beta_j)^2. \tag{46}$$

And then calculate our estimate of the σ^2 of the next iteration with:

$$\sigma^2=\frac{1}{(T-m)}\sum_{j=1}^{N}\sum_{t=m+1}^{T}(y_t^*-(x_t^*)'\beta_j)^2. \tag{47}$$

Therefore, we have now a specific algorithm for estimating the parameters of a Markov-Switching AR model, where all coefficients switch and the error term variance does not switch. This is the case presented in Hamilton (1990, page 56-58). Sadly Hamilton does not show how to apply the EM algorithm to broader structures of AR models, therefore the following examples are the application of the EM algorithm to broader defined AR processes. To the best of the authors knowledge this presentation of applying the EM algorithm to broader defined specific AR processes is novel.

4.4.4 Example 1: Non-Switching Intercept

From the previous derivations, we know that the equations (38), (39), and (40) hold. The five applications of the EM algorithm that we now specifically present are within the class of models for which these three equations were originally derived, i.e. an autoregressive process with a maximum lag order. As opposed to Example 0 we vary the parameters influenced by the underlying Markov-Chain. It is important to emphasize that all models presented are still Markov-Switching AR(m) models with gaussian white noise that fulfill the assumptions made in section 4.1, i.e that fulfill (11), (12), (16) and (17). Assumption (11) is fulfilled because the parameters that describe the generation of Y_t are only directly influenced by the current state of the Markov-Chain s_t and not by earlier states of the Markov-Chain $s_{t-1}, s_{t-2}, ...$, as described in 4.1.

With that general short discussion out of the way we now turn to Example 1. This time we assume that the coefficients switch, while the intercept and the error term variance do not switch. The assumed process therefore has the following form:

$$Y_t = c + \phi_{1,s_t} Y_{t-1} + ... + \phi_{m,s_t} Y_{t-m} + U_t; \quad \text{where} \quad U_t \overset{i.i.d.}{\sim} N(0, \sigma^2).$$

We could alternatively write:

$$Y_t = c + X_t' \phi_{s_t} + U_t \quad \text{with} \quad X_t = \begin{pmatrix} Y_{t-1} \\ Y_{t-2} \\ ... \\ Y_{t-m} \end{pmatrix}, \quad \phi_{s_t} = \begin{pmatrix} \phi_{1,s_t} \\ \phi_{2,s_t} \\ ... \\ \phi_{m,s_t} \end{pmatrix} \quad \text{and} \quad \beta_{s_t} = \begin{pmatrix} c \\ \phi_{1,s_t} \\ \phi_{2,s_t} \\ ... \\ \phi_{m,s_t} \end{pmatrix}.$$

We start again with the conditional log-likelihood, which has the following form:

$$\ln(f_{Y_t|Z_t;\alpha}(y_t|z_t)) = \ln(\frac{1}{\sqrt{2\pi}\sigma}) - \frac{(y_t - c - x_t'\phi_{s_t})^2}{2\sigma^2}.$$

Now we approach this very similar to how we approached Example 0, we take the derivative in α, only that now there is a switching and a non-switching part in β, we have to take the derivative in each. It holds that:

$$\frac{\partial \ln(f_{Y_t|Z_t;\alpha}(y_t|z_t))}{\partial c} = \frac{(y_t - c - x_t'\phi_{s_t})}{\sigma^2} \quad \forall s_t \in \{1, ..., N\}.$$

We substitute our result in (39):

$$\sum_{t=m+1}^{T}\sum_{s_t=1}^{N}\cdots\sum_{s_{t-m}=1}^{N}\frac{(y_t-c_{(l+1)}-x_t'\phi_{s_t}^{(l+1)})}{\sigma_{(l+1)}^2}P_{\lambda_l}(S_t=s_t,...,S_{t-m}=s_{t-m}|\mathscr{Y}_T=\vec{y}_T)=0$$

$$\Leftrightarrow \sum_{t=m+1}^{T}\sum_{s_t=1}^{N}\frac{(y_t-c_{(l+1)}-x_t'\phi_{s_t}^{(l+1)})}{\sigma_{(l+1)}^2}P_{\lambda_l}(S_t=s_t|\mathscr{Y}_T=\vec{y}_T)=0$$

$$\Leftrightarrow \sum_{t=m+1}^{T}\sum_{s_t=1}^{N}(y_t-c_{(l+1)}-x_t'\phi_{s_t}^{(l+1)})P_{\lambda_l}(S_t=s_t|\mathscr{Y}_T=\vec{y}_T)=0$$

$$\Leftrightarrow \sum_{t=m+1}^{T}\sum_{s_t=1}^{N}(y_t-x_t'\phi_{s_t}^{(l+1)})P_{\lambda_l}(S_t=s_t|\mathscr{Y}_T=\vec{y}_T)=\sum_{t=m+1}^{T}\sum_{s_t=1}^{N}c_{(l+1)}P_{\lambda_l}(S_t=s_t|\mathscr{Y}_T=\vec{y}_T)$$

$$\Leftrightarrow \sum_{t=m+1}^{T}\sum_{s_t=1}^{N}(y_t-x_t'\phi_{s_t}^{(l+1)})P_{\lambda_l}(S_t=s_t|\mathscr{Y}_T=\vec{y}_T)=(T-m)c_{(l+1)}$$

$$\Leftrightarrow c_{(l+1)}=\frac{1}{(T-m)}\sum_{t=m+1}^{T}\sum_{s_t=1}^{N}(y_t-x_t'\phi_{s_t}^{(l+1)})P_{\lambda_l}(S_t=s_t|\mathscr{Y}_T=\vec{y}_T).$$

The last result:

$$c_{(l+1)}=\frac{1}{(T-m)}\sum_{t=m+1}^{T}\sum_{j=1}^{N}(y_t-x_t'\phi_{s_t}^{(l+1)})P_{\lambda_l}(S_t=j|\mathscr{Y}_T=\vec{y}_T) \tag{48}$$

can be understood as a constraint. Next, we take the derivative in ϕ_j:

$$\frac{\partial\ln(f_{Y_t|Z_t;\alpha}(y_t|z_t))}{\partial\phi_j}=\frac{(y_t-c-x_t'\phi_j)x_t}{\sigma^2},\quad \text{if } S_t=j, \text{ else } 0.$$

We insert in (39):

$$\sum_{t=m+1}^{T}\sum_{s_t=1}^{N}\cdots\sum_{s_{t-m}=1}^{N}\frac{\partial\ln(f_{Y_t|Z_t;\alpha}(y_t|z_t))}{\partial\phi_j}\Big|_{\alpha=\alpha^{(l+1)}}P_{\lambda_l}(S_t=s_t,...,S_{t-m}=s_{t-m}|\mathscr{Y}_T=\vec{y}_T)=0$$

$$\Leftrightarrow \sum_{t=m+1}^{T}\frac{(y_t-c_{(l+1)}-x_t'\phi_j^{(l+1)})x_t}{\sigma_{(l+1)}^2}P_{\lambda_l}(S_t=j|\mathscr{Y}_T=\vec{y}_T)=0$$

$$\Leftrightarrow \sum_{t=m+1}^{T}(y_t-c_{(l+1)}-x_t'\phi_j^{(l+1)})x_tP_{\lambda_l}(S_t=j|\mathscr{Y}_T=\vec{y}_T)=0.$$

This is equivalent to the FOC of the following optimization problem:

$$\arg\min_{\phi_j}\sum_{t=m+1}^{T}\left((y_t-c)\sqrt{P_{\lambda_l}(S_t=j|\mathscr{Y}_T=\vec{y}_T)}-x_t'\sqrt{P_{\lambda_l}(S_t=j|\mathscr{Y}_T=\vec{y}_T)}\phi_j\right)^2. \tag{49}$$

Finally, it remains to differentiate with respect to σ^2. In this special case, the differentiation proceeds exactly as in Hamilton's example, since σ^2 still does not switch. Thus, we have:

$$\sigma_{(l+1)}^2=\frac{1}{(T-M)}\sum_{t=m+1}^{T}\sum_{j=1}^{N}(y_t-c_{(l+1)}-x_t'\phi_j^{(l+1)})^2P_{\lambda_l}(S_t=j|\mathscr{Y}_T=\vec{y}_T).$$

It thus becomes apparent that σ^2 does not affect the conditions for c and ϕ_j, while these in turn do influence the condition for σ^2. Accordingly, as in Hamilton's example, one can first solve for β_j before determining σ^2. However, what changed is that finding β_j is no longer a simple optimization step, since two conditions must now be satisfied simultaneously—namely, those for ϕ_j and c. It turns

out that simultaneously satisfying both conditions is equivalent to optimizing expression (49) subject to the constraint given by (48). We therefore need to solve the following optimisation problem for β_j:

$$\arg\min_{\phi_j} \sum_{t=m+1}^{T} \left((y_t - c)\sqrt{P_{\lambda_l}(S_t = j|\mathscr{Y}_T = \vec{y}_T)} - x_t'\sqrt{P_{\lambda_l}(S_t = j|\mathscr{Y}_T = \vec{y}_T)}\phi_j \right)^2 \quad s.t.$$

$$c = \frac{1}{(T-m)} \sum_{t=m+1}^{T} \sum_{j=1}^{N} (y_t - x_t'\phi_j) P_{\lambda_l}(S_t = j|\mathscr{Y}_T = \vec{y}_T).$$

Now we can use this β_j, similarly to how we know it from Hamilton and get:

$$\sigma^2 = \frac{1}{(T-M)} \sum_{t=m+1}^{T} \sum_{j=1}^{N} (y_t - c - x_t'\phi_j)^2 P_{\lambda_l}(S_t = j|\mathscr{Y}_T = \vec{y}_T).$$

This concludes our α vector for the next iteration.

4.4.5 Example 2: Switching Intercept and Non-Switching Coefficients

This time, we reverse the roles; instead of letting the coefficients switch, now only the intercept switches. Therefore, the assumed process would have the following form:

$$Y_t = c_{s_t} + \phi_1 Y_{t-1} + ... + \phi_m Y_{t-m} + U_t; \quad \text{where} \quad U_t \overset{i.i.d.}{\sim} N(0, \sigma^2),$$

we could alternatively write:

$$Y_t = c_{s_t} + X_t'\phi + U_t \quad \text{with} \quad X_t = \begin{pmatrix} Y_{t-1} \\ Y_{t-2} \\ ... \\ Y_{t-m} \end{pmatrix}, \quad \phi_{s_t} = \begin{pmatrix} \phi_1 \\ \phi_2 \\ ... \\ \phi_m \end{pmatrix} \quad \text{and} \quad \beta_{s_t} = \begin{pmatrix} c_{s_t} \\ \phi_1 \\ \phi_2 \\ ... \\ \phi_m \end{pmatrix}.$$

In this case, we differentiate with respect to ϕ, c_j, and σ^2. It is important to note that the error term variance still does not switch. For this setup, we can write:

$$\ln(f_{Y_t|Z_t;\alpha}(y_t|z_t)) = \ln(\frac{1}{\sqrt{2\pi}\sigma}) - \frac{(y_t - c_{s_t} - x_t'\phi)^2}{2\sigma^2}.$$

First we differentiate with respect to ϕ and get:

$$\frac{\partial \ln(f_{Y_t|Z_t;\alpha}(y_t|z_t))}{\partial \phi} = \frac{(y_t - c_{s_t} - x_t'\phi)x_t}{\sigma^2} \quad \forall s_t \in \{1, ..., N\}.$$

We insert in (39), this leads us to:

$$\sum_{t=m+1}^{T}\sum_{s_t=1}^{N}\cdots\sum_{s_{t-m}=1}^{N}\frac{(y_t-c_{s_t}^{(l+1)}-x_t'\phi^{(l+1)})x_t}{\sigma^{2(l+1)}}P_{\lambda_l}(S_t=s_t,...,S_{t-m}=s_{t-m}|\mathscr{Y}_T=\vec{y}_T)=0$$

$$\Leftrightarrow \sum_{t=m+1}^{T}\sum_{s_t=1}^{N}\frac{(y_t-c_{s_t}^{(l+1)}-x_t'\phi^{(l+1)})x_t}{\sigma^2_{(l+1)}}P_{\lambda_l}(S_t=s_t|\mathscr{Y}_T=\vec{y}_T)=0$$

$$\Leftrightarrow \sum_{t=m+1}^{T}\sum_{s_t=1}^{N}(y_t-c_{s_t}^{(l+1)}-x_t'\phi^{(l+1)})x_tP_{\lambda_l}(S_t=s_t|\mathscr{Y}_T=\vec{y}_T)=0$$

$$\Leftrightarrow \sum_{t=m+1}^{T}\sum_{s_t=1}^{N}(y_t-c_{s_t}^{(l+1)})x_tP_{\lambda_l}(S_t=s_t|\mathscr{Y}_T=\vec{y}_T)=\sum_{t=m+1}^{T}\sum_{s_t=1}^{N}x_t'\phi^{(l+1)}x_tP_{\lambda_l}(S_t=s_t|\mathscr{Y}_T=\vec{y}_T)$$

$$\Leftrightarrow \sum_{t=m+1}^{T}\sum_{s_t=1}^{N}(y_t-c_{s_t}^{(l+1)})x_tP_{\lambda_l}(S_t=s_t|\mathscr{Y}_T=\vec{y}_T)=\sum_{t=m+1}^{T}\sum_{s_t=1}^{N}P_{\lambda_l}(S_t=s_t|\mathscr{Y}_T=\vec{y}_T)x_tx_t'\phi^{(l+1)}$$

$$\Leftrightarrow \sum_{t=m+1}^{T}\sum_{s_t=1}^{N}(y_t-c_{s_t}^{(l+1)})x_tP_{\lambda_l}(S_t=s_t|\mathscr{Y}_T=\vec{y}_T)=\left(\sum_{t=m+1}^{T}\sum_{s_t=1}^{N}P_{\lambda_l}(S_t=s_t|\mathscr{Y}_T=\vec{y}_T)x_tx_t'\right)\phi^{(l+1)}$$

$$\Leftrightarrow \phi^{(l+1)}=\left(\sum_{t=m+1}^{T}x_tx_t'\right)^{-1}\sum_{t=m+1}^{T}\sum_{s_t=1}^{N}(y_t-c_{s_t}^{(l+1)})x_tP_{\lambda_l}(S_t=s_t|\mathscr{Y}_T=\vec{y}_T).$$

Next we differentiate with respect to c_j:

$$\frac{\partial\ln(f_{Y_t|Z_t;\alpha}(y_t|z_t))}{\partial c_j}=\frac{(y_t-c_j-x_t'\phi)}{\sigma^2}\quad\text{if } S_t=j\text{, else } 0.$$

Similarily we insert the result in (39):

$$\sum_{t=m+1}^{T}\sum_{s_t=1}^{N}\cdots\sum_{s_{t-m}=1}^{N}\frac{\partial\ln(f_{Y_t|Z_t;\alpha}(y_t|z_t))}{\partial c_j}\Big|_{\alpha=\alpha^{(l+1)}}P_{\lambda_l}(S_t=s_t,...,S_{t-m}=s_{t-m}|\mathscr{Y}_T=\vec{y}_T)=0$$

$$\Leftrightarrow \sum_{t=m+1}^{T}\frac{(y_t-c_j^{(l+1)}-x_t'\phi^{(l+1)})}{\sigma^2_{(l+1)}}P_{\lambda_l}(S_t=j|\mathscr{Y}_T=\vec{y}_T)=0$$

$$\Leftrightarrow \sum_{t=m+1}^{T}(y_t-c_j^{(l+1)}-x_t'\phi^{(l+1)})P_{\lambda_l}(S_t=j|\mathscr{Y}_T=\vec{y}_T)=0.$$

It should be noted that the last equation is the FOC of the following optimization problem:

$$\arg\min_{c_j}\sum_{t=m+1}^{T}\left((y_t-x_t'\phi)\sqrt{P_{\lambda_l}(S_t=j|\mathscr{Y}_T=\vec{y}_T)}-c_j\sqrt{P_{\lambda_l}(S_t=j|\mathscr{Y}_T=\vec{y}_T)}\right)^2. \tag{50}$$

For σ^2, the same condition applies as already formulated by Hamilton, because σ^2 again does not switch. Thus, we are in a very similar situation as in the previous example, because the conditions resulting from the derivative with respect to c_j, as well as the condition resulting from the derivative with respect to ϕ, must be satisfied simultaneously and affect the condition for σ^2, whereas σ^2 does not affect the former conditions. Therefore, one can first satisfy the first two conditions for all j in order to obtain β_j for all j, and then determine σ^2 for the next iteration. Furthermore, it follows again that β_j, for a given j, can be found by solving an optimization problem with an equality constraint. In

order to obtain β_j, the following optimization problem must be solved:

$$\arg\min_{c_j} \sum_{t=m+1}^{T} \left((y_t - x_t'\phi)\sqrt{P_{\lambda_l}(S_t = j|\mathscr{Y}_T = \vec{y}_T)} - c_j\sqrt{P_{\lambda_l}(S_t = j|\mathscr{Y}_T = \vec{y}_T)} \right)^2 \quad s.t.$$

$$\phi = \left(\sum_{t=m+1}^{T} x_t x_t' \right)^{-1} \sum_{t=m+1}^{T} \sum_{j=1}^{N} (y_t - c_j) x_t P_{\lambda_l}(S_t = j|\mathscr{Y}_T = \vec{y}_T).$$

As usual we compute:

$$\sigma^2 = \frac{1}{(T-M)} \sum_{t=m+1}^{T} \sum_{j=1}^{N} (y_t - c - x_t'\phi_j)^2 P_{\lambda_l}(S_t = j|\mathscr{Y}_T = \vec{y}_T),$$

and with that we get out α vector for the next iteration.

4.4.6 Example 3: All Parameters Switch

As a third example, we now consider the case in which all parameters are allowed to switch. That is, we now allow not only the coefficients and the intercept to switch, but also the variance of the error term. It is important to note that in setups discussed earlier one could have made stronger assumptions, namely it would have been easier to assume in Example 0 to Example 2, that S_t was independent of U_τ for all t and τ. Instead we made the weaker assumptions (16) and (17) so that we can now introduce models where the error term variance is allowed to switch, that wouldn't have been possible with the stronger set of assumptions. That is the reason why all derivations earlier were done with this weaker set of assumptions. That said, we want to point out that we still make the same assumptions as in section 4.1, this is possible due to the weaker set of assumptions, a stronger set of assumptions wouldn't allow for models like Example 3 and Example 5. Additionally, it should be noted that, in terms of notation, we now again include a 1 as the first element of X_t to represent the intercept. This leads to the following process formulation:

$$y_t = c_{s_t} + \phi_{1,s_t} Y_{t-1} + ... + \phi_{m,s_t} Y_{t-m} + U_t; \quad \text{where} \quad U_t \sim N(0, \sigma_{s_t}^2),$$

we could alternatively write:

$$Y_t = X_t'\beta_{s_t} + U_t \quad \text{with} \quad X_t = \begin{pmatrix} 1 \\ Y_{t-1} \\ Y_{t-2} \\ ... \\ Y_{t-m} \end{pmatrix} \quad \text{and} \quad \beta_{s_t} = \begin{pmatrix} c_{s_t} \\ \phi_{1,s_t} \\ \phi_{2,s_t} \\ ... \\ \phi_{m,s_t} \end{pmatrix}.$$

As usual we start with the conditional log-likelihood:

$$\ln(f_{Y_t|Z_t;\alpha}(y_t|z_t)) = \ln(\frac{1}{\sqrt{2\pi}\sigma_{s_t}}) - \frac{(y_t - x_t\beta_{s_t})^2}{2\sigma_{s_t}^2}.$$

We now need to take the derivative once with respect to β_j and once with respect to σ_j^2 for a given j. If we first take the derivative with respect to β_j, we obtain:

$$\frac{\partial \ln(f_{Y_t|Z_t;\alpha}(y_t|z_t))}{\partial \beta_j} = \frac{(y_t - x_t'\beta_j)x_t}{\sigma_j^2} \quad \text{if } S_t = j, \text{ else } 0.$$

We now substitute this into (39) and obtain:

$$\sum_{t=m+1}^{T}\sum_{s_t=1}^{N}\cdots\sum_{s_{t-m}=1}^{N}\frac{\partial \ln(f_{Y_t|Z_t;\alpha}(y_t|z_t))}{\partial\beta_j}\Big|_{\alpha=\alpha^{(l+1)}}P_{\lambda_l}(S_t=s_t,...,S_{t-m}=s_{t-m}|\mathscr{Y}_T=\vec{y}_T)=0$$

$$\Leftrightarrow \sum_{t=m+1}^{T}\frac{(y_t-x_t'\beta_j^{(l+1)})x_t}{\sigma_{j,(l+1)}^2}P_{\lambda_l}(S_t=j|\mathscr{Y}_T=\vec{y}_T)=0.$$

We again observe that this corresponds to the FOC of an optimization problem, namely the following optimization problem:

$$\arg\min_{\beta_j}\sum_{t=m+1}^{T}\left(\frac{y_t}{\sigma_j}\sqrt{P_{\lambda_l}(S_t=j|\mathscr{Y}_T=\vec{y}_T)}-\frac{x_t'}{\sigma_j}\sqrt{P_{\lambda_l}(S_t=j|\mathscr{Y}_T=\vec{y}_T)}\beta_j\right)^2.$$

Next we take the derivative with respect to σ_j^{-2} and get:

$$\frac{\partial \ln(f_{Y_t|Z_t;\alpha}(y_t|z_t))}{\partial\sigma_j^{-2}}=\frac{\sigma_j^2}{2}-\frac{(y_t-x_t'\beta_j)^2}{2}\quad\text{if } S_t=j\text{, else 0.}$$

We substitute the result into (39), this leads to:

$$\sum_{t=m+1}^{T}\sum_{s_t=1}^{N}\cdots\sum_{s_{t-m}=1}^{N}\frac{\partial \ln(f_{Y_t|Z_t;\alpha}(y_t|z_t))}{\partial\sigma_j^{-2}}\Big|_{\alpha=\alpha^{(l+1)}}P_{\lambda_l}(S_t=s_t,...,S_{t-m}=s_{t-m}|\mathscr{Y}_T=\vec{y}_T)=0$$

$$\Leftrightarrow \sum_{t=m+1}^{T}\left(\frac{\sigma_{j,(l+1)}^2}{2}-\frac{(y_t-x_t'\beta_j^{(l+1)})^2}{2}\right)P_{\lambda_l}(S_t=j|\mathscr{Y}_T=\vec{y}_T)=0$$

$$\Leftrightarrow \sum_{t=m+1}^{T}(\sigma_{j,(l+1)}^2-(y_t-x_t'\beta_j^{(l+1)})^2)P_{\lambda_l}(S_t=j|\mathscr{Y}_T=\vec{y}_T)=0$$

$$\Leftrightarrow \sum_{t=m+1}^{T}\sigma_{j,(l+1)}^2P_{\lambda_l}(S_t=j|\mathscr{Y}_T=\vec{y}_T)=\sum_{t=m+1}^{T}(y_t-x_t'\beta_j^{(l+1)})^2P_{\lambda_l}(S_t=j|\mathscr{Y}_T=\vec{y}_T)$$

$$\Leftrightarrow \sigma_{j,(l+1)}^2\sum_{t=m+1}^{T}P_{\lambda_l}(S_t=j|\mathscr{Y}_T=\vec{y}_T)=\sum_{t=m+1}^{T}(y_t-x_t'\beta_j^{(l+1)})^2P_{\lambda_l}(S_t=j|\mathscr{Y}_T=\vec{y}_T)$$

$$\Leftrightarrow \sigma_{j,(l+1)}^2=\frac{\sum_{t=m+1}^{T}(y_t-x_t'\beta_j^{(l+1)})^2P_{\lambda_l}(S_t=j|\mathscr{Y}_T=\vec{y}_T)}{\sum_{t=m+1}^{T}P_{\lambda_l}(S_t=j|\mathscr{Y}_T=\vec{y}_T)}.$$

As a result, since σ_j now affects the condition for β_j and vice versa, we once again arrive at a constrained optimization problem, which leads to α of the next iteration. The constrained optimization problem is given by:

$$\arg\min_{\beta_j}\sum_{t=m+1}^{T}\left(\frac{y_t}{\sigma_j}\sqrt{P_{\lambda_l}(S_t=j|\mathscr{Y}_T=\vec{y}_T)}-\frac{x_t'}{\sigma_j}\sqrt{P_{\lambda_l}(S_t=j|\mathscr{Y}_T=\vec{y}_T)}\beta_j\right)^2\quad s.t.$$

$$\sigma_j=\sqrt{\frac{\sum_{t=m+1}^{T}(y_t-x_t'\beta_j)^2P_{\lambda_l}(S_t=j|\mathscr{Y}_T=\vec{y}_T)}{\sum_{t=m+1}^{T}P_{\lambda_l}(S_t=j|\mathscr{Y}_T=\vec{y}_T)}}.$$

4.4.7 Example 4: Arbitrary Subset-Switching of (c,ϕ) and Non-Switching σ^2

The three previous examples are essentially special cases of the two model formulations that follow. Therefore, the next two examples represent the most general forms of Markov-Switching AR(m) models that will appear in this paper. The model class introduced next assumes that σ^2 does not switch,

and that an arbitrary subset of β is allowed to switch. Accordingly, we divide the parameter vector β into the switching components, denoted by β^S, and the non-switching components, denoted by β^F. The underlying process is therefore formulated as follows:

$$Y_t = (X_t^F)'\beta^F + (X_t^S)'\beta_{s_t}^S + U_t; \quad \text{where} \quad U_t \overset{i.i.d.}{\sim} N(0,\sigma^2) \quad \text{and} \quad X_t = \begin{pmatrix} 1 \\ Y_{t-1} \\ Y_{t-2} \\ \dots \\ Y_{t-m} \end{pmatrix}.$$

Thereby X_t^F and X_t^S are defined such that their elements do not overlap and that the elements of both vectors together are the elements of X_t, to put it more formally: Let $I^F, I^S \subset \{1,\dots,m+1\}$ be disjoint index sets such that $I^F \cap I^S = \emptyset$ and $I^F \cup I^S = \{1,\dots,m+1\}$, where $m+1$ is the number of coefficients plus intercept. Then we define

$$X_t^F = ((X_t)_i)_{i\in I^F}, \quad X_t^S = ((X_t)_i)_{i\in I^S}.$$

Again we start with the conditional log-likelihood:

$$\ln(f_{Y_t|Z_t;\alpha}(y_t|z_t)) = \ln(\frac{1}{\sqrt{2\pi}\sigma}) - \frac{(y_t - (x_t^S)'\beta_{s_t}^S - (x_t^F)'\beta^F)^2}{2\sigma^2}.$$

Accordingly, for this model class, we need to take the derivative with respect to σ^2, β_j^S, and β^F. We will start with β_j^S:

$$\frac{\partial \ln(f_{Y_t|Z_t;\alpha}(y_t|z_t))}{\partial \beta_j^S} = \frac{(y_t - (x_t^S)'\beta_j^S - (x_t^F)'\beta^F)x_t^S}{\sigma^2} \quad \text{if } S_t = j, \text{ else } 0.$$

We can now substitute this into (39) and obtain:

$$\sum_{t=m+1}^{T}\sum_{s_t=1}^{N}\cdots\sum_{s_{t-m}=1}^{N}\frac{\partial \ln(f_{Y_t|Z_t;\alpha}(y_t|z_t))}{\partial \beta_j^S}\Big|_{\alpha=\alpha^{(l+1)}} P_{\lambda_l}(S_t = s_t,\dots,S_{t-m} = s_{t-m}|\mathscr{Y}_T = \vec{y}_T) = 0$$

$$\Leftrightarrow \sum_{t=m+1}^{T}(y_t - (x_t^S)'\beta_{j,(l+1)}^S - (x_t^F)'\beta_{(l+1)}^F)x_t^S P_{\lambda_l}(S_t = j|\mathscr{Y}_T = \vec{y}_T) = 0.$$

This, in turn, corresponds to the FOC of the following optimization problem:

$$\arg\min_{\beta_j^S}\sum_{t=m+1}^{T}\left((y_t - (x_t^F)'\beta^F)\sqrt{P_{\lambda_l}(S_t = j|\mathscr{Y}_T = \vec{y}_T)} - (x_t^S)'\sqrt{P_{\lambda_l}(S_t = j|\mathscr{Y}_T = \vec{y}_T)}\beta_j^S\right)^2.$$

Next, if we take the derivative with respect to β^F, we obtain:

$$\frac{\partial \ln(f_{Y_t|Z_t;\alpha}(y_t|z_t))}{\partial \beta^F} = \frac{(y_t - (x_t^S)'\beta_j^S - (x_t^F)'\beta^F)x_t^F}{\sigma^2} \quad \forall s_t \in \{1,\dots,N\}.$$

We substitute this into (39) and obtain:

$$\sum_{t=m+1}^{T}\sum_{s_t=1}^{N}\cdots\sum_{s_{t-m}=1}^{N}\frac{(y_t-(x_t^S)'\beta^S_{s_t,(l+1)}-(x_t^F)'\beta^F_{(l+1)})x_t^F}{\sigma^2_{(l+1)}}P_{\lambda_l}(S_t=s_t,...,S_{t-m}=s_{t-m}|\mathscr{Y}_T=\vec{y}_T)=0$$

$$\Leftrightarrow \sum_{t=m+1}^{T}\sum_{s_t=1}^{N}(y_t-(x_t^S)'\beta^S_{s_t,(l+1)}-(x_t^F)'\beta^F_{(l+1)})x_t^F P_{\lambda_l}(S_t=s_t|\mathscr{Y}_T=\vec{y}_T)=0$$

$$\Leftrightarrow \sum_{t=m+1}^{T}\sum_{s_t=1}^{N}(y_t-(x_t^S)'\beta^S_{s_t,(l+1)})x_t^F P_{\lambda_l}(S_t=s_t|\mathscr{Y}_T=\vec{y}_T)=\sum_{t=m+1}^{T}\sum_{s_t=1}^{N}(x_t^F)'\beta^F_{(l+1)}x_t^F P_{\lambda_l}(S_t=s_t|\mathscr{Y}_T=\vec{y}_T)$$

$$\Leftrightarrow \sum_{t=m+1}^{T}\sum_{s_t=1}^{N}(y_t-(x_t^S)'\beta^S_{s_t,(l+1)})x_t^F P_{\lambda_l}(S_t=s_t|\mathscr{Y}_T=\vec{y}_T)=\sum_{t=m+1}^{T}(x_t^F)'\beta^F_{(l+1)}x_t^F$$

$$\Leftrightarrow \sum_{t=m+1}^{T}\sum_{s_t=1}^{N}(y_t-(x_t^S)'\beta^S_{s_t,(l+1)})x_t^F P_{\lambda_l}(S_t=s_t|\mathscr{Y}_T=\vec{y}_T)=\sum_{t=m+1}^{T}x_t^F(x_t^F)'\beta^F_{(l+1)}$$

$$\Leftrightarrow \sum_{t=m+1}^{T}\sum_{s_t=1}^{N}(y_t-(x_t^S)'\beta^S_{s_t,(l+1)})x_t^F P_{\lambda_l}(s_t=s_t|\mathscr{Y}_T=\vec{y}_T)=\left(\sum_{t=m+1}^{T}x_t^F(x_t^F)'\right)\beta^F_{(l+1)}$$

$$\Leftrightarrow \beta^F_{(l+1)}=\left(\sum_{t=m+1}^{T}x_t^F(x_t^F)'\right)^{-1}\left(\sum_{t=m+1}^{T}\sum_{s_t=1}^{N}(y_t-(x_t^S)'\beta^S_{s_t,(l+1)})x_t^F P_{\lambda_l}(S_t=s_t|\mathscr{Y}_T=\vec{y}_T)\right).$$

For this generalized model class, where arbitrary parameters can switch except for the error term variance, which can not switch, we thus obtain the following constrained optimization problem to determine β^F and β^S_j for all j.

$$\arg\min_{\beta^S_j}\sum_{t=m+1}^{T}\left((y_t-(x_t^F)'\beta^F)\sqrt{P_{\lambda_l}(S_t=j|\mathscr{Y}_T=\vec{y}_T)}-(x_t^S)'\sqrt{P_{\lambda_l}(S_t=j|\mathscr{Y}_T=\vec{y}_T)}\beta^S_j\right)^2 \quad s.t.$$

$$\beta^F=\left(\sum_{t=m+1}^{T}x_t^F(x_t^F)'\right)^{-1}\left(\sum_{t=m+1}^{T}\sum_{j=1}^{N}(y_t-(x_t^S)'\beta^S_j)x_t^F P_{\lambda_l}(S_t=j|\mathscr{Y}_T=\vec{y}_T)\right).$$

We then calculate, as usual:

$$\sigma^2=\frac{1}{(T-M)}\sum_{t=m+1}^{T}\sum_{j=1}^{N}(y_t-x_t'\beta_j)^2P_{\lambda_l}(S_t=j|\mathscr{Y}_T=\vec{y}_T).$$

and thus obtain our new α vector for the next iteration.

4.4.8 Example 5: Arbitrary Subset-Switching of (c,ϕ) and Switching σ^2

With Example 5, we complete the generalization of the application of the EM algorithm to underlying AR(m) models, as Example 4 and Example 5 together allow for selecting any arbitrary subset of parameters in an AR(m) context for switching. In this final example, we assume the following underlying process:

$$Y_t=(X_t^F)'\beta^F+(X_t^S)'\beta^S_{s_t}+U_t;\quad \text{where}\quad U_t\sim N(0,\sigma^2_{s_t})\quad\text{and}\quad X_t=\begin{pmatrix}1\\Y_{t-1}\\Y_{t-2}\\...\\Y_{t-m}\end{pmatrix}.$$

Thereby, X_t^F and X_t^S are defined such that their elements do not overlap and that the elements of both vectors together are the elements of X_t, to put it more formally: Let $I^F, I^S \subset \{1,\dots,m+1\}$ be disjoint index sets such that $I^F \cap I^S = \emptyset$ and $I^F \cup I^S = \{1,\dots,m+1\}$, where $m+1$ is the number of coefficients plus intercept. Then we define

$$X_t^F = ((X_t)_i)_{i\in I^F}, \quad X_t^S = ((X_t)_i)_{i\in I^S}.$$

Again, we start with the conditional log-liklihood, which would be in this case:

$$\ln(f_{Y_t|Z_t;\alpha}(y_t|z_t)) = \ln\left(\frac{1}{\sqrt{2\pi}\sigma_{s_t}}\right) - \frac{(y_t - (x_t^S)'\beta_{s_t}^S - (x_t^F)'\beta^F)^2}{2\sigma_{s_t}^2}.$$

Accordingly, we need to take the derivative with respect to σ_j^2, β_j^S, and β^F. We begin with the derivative with respect to β_j^S.

$$\frac{\partial \ln(f_{Y_t|Z_t;\alpha}(y_t|z_t))}{\partial \beta_j^S} = \frac{(y_t - (x_t^S)'\beta_j^S - (x_t^F)'\beta^F)x_t^S}{\sigma_j^2} \quad \text{if } S_t = j, \text{ else } 0.$$

We now substitute this into (39) and obtain:

$$\sum_{t=m+1}^{T}\sum_{s_t=1}^{N}\cdots\sum_{s_{t-m}=1}^{N}\frac{\partial \ln(f_{Y_t|Z_t;\alpha}(y_t|z_t))}{\partial \beta_j^S}\Big|_{\alpha=\alpha^{(l+1)}} P_{\lambda_l}(S_t = s_t,\dots,S_{t-m} = s_{t-m}|\mathscr{Y}_T = \vec{y}_T) = 0$$

$$\Leftrightarrow \sum_{t=m+1}^{T}\frac{(y_t - (x_t^S)'\beta_{j,(l+1)}^S - (x_t^F)'\beta_{(l+1)}^F)x_t^S}{\sigma_{j,(l+1)}^2} P_{\lambda_l}(S_t = j|\mathscr{Y}_T = \vec{y}_T) = 0.$$

This, in turn, corresponds to the FOC of the following optimization problem:

$$\arg\min_{\beta_j^S}\sum_{t=m+1}^{T}\left((y_t - (x_t^F)'\beta^F)\frac{\sqrt{P_{\lambda_l}(S_t = j|\mathscr{Y}_T = \vec{y}_T)}}{\sigma_j} - (x_t^S)'\frac{\sqrt{P_{\lambda_l}(S_t = j|\mathscr{Y}_T = \vec{y}_T)}}{\sigma_j}\beta_j^S\right)^2.$$

We now move on to the next step and take the derivative with respect to β^F, obtaining:

$$\frac{\partial \ln(f_{Y_t|Z_t;\alpha}(y_t|z_t))}{\partial \beta^F} = \frac{(y_t - (x_t^S)'\beta_{s_t}^S - (x_t^F)'\beta^F)x_t^F}{\sigma_{s_t}^2} \quad \forall s_t \in \{1,\dots,N\}.$$

We then substitute this into (39) and obtain:

$$\sum_{t=m+1}^{T}\sum_{s_t=1}^{N}\cdots\sum_{s_{t-m}=1}^{N}\frac{(y_t-(x_t^S)'\beta^S_{s_t,(l+1)}-(x_t^F)'\beta^F_{(l+1)})x_t^F}{\sigma^2_{s_t,(l+1)}}P_{\lambda_l}(S_t=s_t,...,S_{t-m}=s_{t-m}|\mathscr{Y}_T=\vec{y}_T)=0$$

$$\Leftrightarrow\sum_{t=m+1}^{T}\sum_{s_t=1}^{N}\frac{(y_t-(x_t^S)'\beta^S_{s_t,(l+1)}-(x_t^F)'\beta^F_{(l+1)})x_t^F}{\sigma^2_{s_t,(l+1)}}P_{\lambda_l}(S_t=s_t|\mathscr{Y}_T=\vec{y}_T)=0$$

$$\Leftrightarrow\sum_{t=m+1}^{T}\sum_{s_t=1}^{N}(y_t-(x_t^S)'\beta^S_{s_t,(l+1)})x_t^F\frac{P_{\lambda_l}(S_t=s_t|\mathscr{Y}_T=\vec{y}_T)}{\sigma^2_{s_t,(l+1)}}=\sum_{t=m+1}^{T}\sum_{s_t=1}^{N}(x_t^F)'\beta^F_{(l+1)}x_t^F\frac{P_{\lambda_l}(S_t=s_t|\mathscr{Y}_T=\vec{y}_T)}{\sigma^2_{s_t,(l+1)}}$$

$$\Leftrightarrow\sum_{t=m+1}^{T}\sum_{s_t=1}^{N}(y_t-(x_t^S)'\beta^S_{s_t,(l+1)})x_t^F\frac{P_{\lambda_l}(S_t=s_t|\mathscr{Y}_T=\vec{y}_T)}{\sigma^2_{s_t,(l+1)}}=\sum_{t=m+1}^{T}(x_t^F)'\beta^F_{(l+1)}x_t^F\sum_{s_t=1}^{N}\frac{P_{\lambda_l}(S_t=s_t|\mathscr{Y}_T=\vec{y}_T)}{\sigma^2_{s_t,(l+1)}}$$

$$\Leftrightarrow\sum_{t=m+1}^{T}\sum_{s_t=1}^{N}(y_t-(x_t^S)'\beta^S_{s_t,(l+1)})x_t^F\frac{P_{\lambda_l}(S_t=s_t|\mathscr{Y}_T=\vec{y}_T)}{\sigma^2_{s_t,(l+1)}}=\sum_{t=m+1}^{T}x_t^F(x_t^F)'\beta^F_{(l+1)}\sum_{s_t=1}^{N}\frac{P_{\lambda_l}(S_t=s_t|\mathscr{Y}_T=\vec{y}_T)}{\sigma^2_{s_t,(l+1)}}$$

$$\Leftrightarrow\sum_{t=m+1}^{T}\sum_{s_t=1}^{N}(y_t-(x_t^S)'\beta^S_{s_t,(l+1)})x_t^F\frac{P_{\lambda_l}(S_t=s_t|\mathscr{Y}_T=\vec{y}_T)}{\sigma^2_{s_t,(l+1)}}=\sum_{t=m+1}^{T}x_t^F(x_t^F)'\sum_{s_t=1}^{N}\frac{P_{\lambda_l}(S_t=s_t|\mathscr{Y}_T=\vec{y}_T)}{\sigma^2_{s_t,(l+1)}}\beta^F_{(l+1)}$$

$$\Leftrightarrow\sum_{t=m+1}^{T}\sum_{s_t=1}^{N}(y_t-(x_t^S)'\beta^S_{s_t,(l+1)})x_t^F\frac{P_{\lambda_l}(S_t=s_t|\mathscr{Y}_T=\vec{y}_T)}{\sigma^2_{s_t,(l+1)}}=\left(\sum_{t=m+1}^{T}x_t^F(x_t^F)'\sum_{s_t=1}^{N}\frac{P_{\lambda_l}(S_t=s_t|\mathscr{Y}_T=\vec{y}_T)}{\sigma^2_{s_t,(l+1)}}\right)\beta^F_{(l+1)}$$

$$\Leftrightarrow\beta^F_{(l+1)}=\left(\sum_{t=m+1}^{T}x_t^F(x_t^F)'\sum_{s_t=1}^{N}\frac{P_{\lambda_l}(S_t=s_t|\mathscr{Y}_T=\vec{y}_T)}{\sigma^2_{s_t,(l+1)}}\right)^{-1}\left(\sum_{t=m+1}^{T}\sum_{s_t=1}^{N}(y_t-(x_t^S)'\beta^S_{s_t,(l+1)})x_t^F\frac{P_{\lambda_l}(S_t=s_t|\mathscr{Y}_T=\vec{y}_T)}{\sigma^2_{s_t,(l+1)}}\right).$$

As a third step, we now take the derivative with respect to σ_j^{-2} and obtain:

$$\frac{\partial\ln(f_{Y_t|Z_t;\alpha}(y_t|z_t))}{\partial\sigma_j^{-2}}=\frac{\sigma_j^2}{2}-\frac{(y_t-(x_t^S)'\beta_j^S-(x_t^F)'\beta^F)^2}{2}\quad\text{if } S_t=j \text{ else } 0.$$

We now substitute this into (39) and obtain:

$$\sum_{t=m+1}^{T}\sum_{s_t=1}^{N}\cdots\sum_{s_{t-m}=1}^{N}\frac{\partial\ln(f_{Y_t|Z_t;\alpha}(y_t|z_t))}{\partial\sigma_j^{-2}}\Big|_{\alpha=\alpha^{(l+1)}}P_{\lambda_l}(S_t=s_t,...,S_{t-m}=s_{t-m}|\mathscr{Y}_T=\vec{y}_T)=0$$

$$\Leftrightarrow\sum_{t=m+1}^{T}\left(\frac{\sigma^2_{j,(l+1)}}{2}-\frac{(y_t-(x_t^S)'\beta^S_{j,(l+1)}-(x_t^F)'\beta^F_{(l+1)})^2}{2}\right)P_{\lambda_l}(S_t=j|\mathscr{Y}_T=\vec{y}_T)=0$$

$$\Leftrightarrow\sum_{t=m+1}^{T}\sigma^2_{j,(l+1)}P_{\lambda_l}(S_t=j|\mathscr{Y}_T=\vec{y}_T)=\sum_{t=m+1}^{T}(y_t-(x_t^S)'\beta^S_{j,(l+1)}-(x_t^F)'\beta^F_{(l+1)})^2P_{\lambda_l}(S_t=j|\mathscr{Y}_T=\vec{y}_T)$$

$$\Leftrightarrow\sigma^2_{j,(l+1)}\sum_{t=m+1}^{T}P_{\lambda_l}(S_t=j|\mathscr{Y}_T=\vec{y}_T)=\sum_{t=m+1}^{T}(y_t-(x_t^S)'\beta^S_{j,s_t}-(x_t^F)'\beta^F_{(l+1)})^2P_{\lambda_l}(S_t=j|\mathscr{Y}_T=\vec{y}_T)$$

$$\Leftrightarrow\sigma^2_{j,(l+1)}=\frac{\sum_{t=m+1}^{T}(y_t-(x_t^S)'\beta^S_{j,s_t}-(x_t^F)'\beta^F_{(l+1)})^2P_{\lambda_l}(S_t=j|\mathscr{Y}_T=\vec{y}_T)}{\sum_{t=m+1}^{T}P_{\lambda_l}(S_t=j|\mathscr{Y}_T=\vec{y}_T)}.$$

We can now combine these three results into a single optimization problem, which leads us to the next α. This is necessary because, once again, all three conditions must be satisfied simultaneously and cannot be implemented sequentially, as each of the three variables plays a role in the different condi-

tions. We thus obtain another constrained optimization problem, but this time under two constraints:

$$\arg\min_{\beta_j^S} \sum_{t=m+1}^{T} \left((y_t - (x_t^F)'\beta^F) \frac{\sqrt{P_{\lambda_l}(S_t = j|\mathscr{Y}_T = \vec{y}_T)}}{\sigma_j} - (x_t^S)' \frac{\sqrt{P_{\lambda_l}(S_t = j|\mathscr{Y}_T = \vec{y}_T)}}{\sigma_j} \beta_j^S \right)^2 \quad s.t.$$

$$\beta^F = \left(\sum_{t=m+1}^{T} x_t^F (x_t^F)' \sum_{j=1}^{N} \frac{P_{\lambda_l}(S_t = j|\mathscr{Y}_T = \vec{y}_T)}{\sigma_j^2} \right)^{-1} \left(\sum_{t=m+1}^{T} \sum_{j=1}^{N} (y_t - (x_t^S)'\beta_j^S) x_t^F \frac{P_{\lambda_l}(S_t = j|\mathscr{Y}_T = \vec{y}_T)}{\sigma_j^2} \right)$$

$$\sigma_j = \sqrt{\frac{\sum_{t=m+1}^{T} (y_t - (x_t^S)'\beta_j^S - (x_t^F)'\beta^F)^2 P_{\lambda_l}(S_t = j|\mathscr{Y}_T = \vec{y}_T)}{\sum_{t=m+1}^{T} P_{\lambda_l}(s_t = j|\mathscr{Y}_T = \vec{y}_T)}}.$$

4.5 Forecasting with Markov-Switching Models

Next, we would like to turn to the topic of forecasts using Markov-Switching models. We recall that the conditional density is given by $f_{Y_t|S_t,\mathscr{Y}_{t-1};\alpha}(y_t|j,\vec{y}_{t-1})$. If we now have $\vec{y}_t$ and s_{t+1}, we could, for a simple AR(1) model, state:

$$Y_{t+1} = c_{s_{t+1}} + \phi_{s_{t+1}} Y_t + U_{t+1},$$

where we have $E_\alpha(Y_{t+1}|S_{t+1} = j, \mathscr{Y}_t = \vec{y}_t) = c_j + \phi_j y_t$. Furthermore, we can show the following for an m-step ahead forecast:

$$\begin{aligned}
E_\theta(Y_{t+m}|\mathscr{Y}_t = \vec{y}_t) &= \int y_{t+m} f_{Y_{t+m}|\mathscr{Y}_t;\theta}(y_{t+m}|\vec{y}_t)\, dy_{t+m} \\
&= \int y_{t+m} \left(\sum_{j=1}^{N} f_{Y_{t+m},S_{t+m}|\mathscr{Y}_T;\theta}(y_{t+m}, j|\vec{y}_t) \right) dy_{t+m} \\
&= \int y_{t+m} \left(\sum_{j=1}^{N} f_{Y_{t+1}|S_{t+m},\mathscr{Y}_t;\alpha}(y_{t+m}|j,\vec{y}_t) P_\theta(S_{t+m} = j|\mathscr{Y}_t = \vec{y}_t) \right) dy_{t+m} \\
&= \sum_{j=1}^{N} P_\theta(S_{t+m} = j|\mathscr{Y}_t = \vec{y}_t) \int y_{t+m} f_{Y_{t+m}|S_{t+m},\mathscr{Y}_t;\alpha}(y_{t+m}|j,\vec{y}_t)\, dy_{t+m} \\
&= \sum_{j=1}^{N} P_\theta(S_{t+m} = j|\mathscr{Y}_t = \vec{y}_t) E_\alpha(Y_{t+m}|S_{t+m} = j, \mathscr{Y}_t = \vec{y}_t).
\end{aligned}$$

One could thus say that the forecasts for y_{t+m} correspond to a weighted average of the expected values given the regime, with the regime probabilities as the weights. To summarize this notation, we can say that we collect the $E_\alpha(Y_{t+m}|S_{t+m} = j, \mathscr{Y}_t = \vec{y}_t)$ in $\mathbf{h}_\mathbf{t}'$, so that we can write $E_\theta(Y_{t+m}|\mathscr{Y}_t = \vec{y}_t) = \mathbf{h}_\mathbf{t}' \hat{\vec{\zeta}}_{t+m|t}$, this derivation closely followed Hamilton (1994, page 694-695).

4.6 Regime Forecasting with Markov-Switching Models

The probability that the Markov-Chain will be in a particular state in the future can be considered as $E_\theta(\zeta_{t+m}|\mathscr{Y}_t = \vec{y}_t)$. Based on (14), this is given by:

$$E_\theta(\zeta_{t+m}|\mathscr{Y}_t = \vec{y}_t) = (\Pi')^m E_\theta(\zeta_t|\mathscr{Y}_t = \vec{y}_t), \tag{51}$$

or alternatively:

$$\hat{\zeta}_{t+m|t} = (\Pi')^m \hat{\zeta}_{t|t}. \tag{52}$$

5 MSwM - The current R standard

Due to the practical relevance of Markov-Switching AR models there are already existing packages for estimating them in R. Thereby the package MSwM[2] is one of the most popular packages. Generally speaking, MSwM should be the first package one finds when searching for R packages regarding Markov-Switching AR models and is often discussed in blog posts online, see for example Lee (2022). One of the advantages of this package is that it allows for specifying which parameters should switch and which should not, via a simple TRUE/FALSE vector. Furthermore, the source code for the package is publicly available via GitHub[3], here we can find the functions ".MSM.em" and ".MSM.lm.maximEM", which are essential for the implementation of the EM algorithm for MSwM, see Sanchez-Espigares & Lopez-Moreno (2021, lines 1154-1170) and Sanchez-Espigares & Lopez-Moreno (2021, lines 1216-1293). As one can observe from this code, the MSwM package utilizes a "stacked-matrix" approach. The following example provides a more detailed explanation of the transformation used. Let us assume we have an AR model with two lags Y_{t-1}, Y_{t-2}, where the coefficient of Y_{t-1} switches and the coefficient of Y_{t-2} is fixed. Furthermore, there shall only be two underlying regimes. The model would then look like this:

$$Y_t = \beta_{1,s_t} Y_{t-1} + \beta_2 Y_{t-2} + U_t,$$

the MSwM package would then transform, for four observations $(y_t, y_{t-1}, y_{t-2}, y_{t-4})$, the data/observations such that we get the following matrices:

$$\tilde{y} = \begin{pmatrix} y_{t-1} \\ y_{t-1} \\ y_t \\ y_t \end{pmatrix}, \quad \tilde{x} = \begin{pmatrix} y_{t-2} & 0 & y_{t-3} \\ 0 & y_{t-2} & y_{t-3} \\ y_{t-1} & 0 & y_{t-2} \\ 0 & y_{t-1} & y_{t-2} \end{pmatrix}, \quad \tilde{w} = \begin{pmatrix} p_{t-1,1} \\ p_{t-1,2} \\ p_{t,1} \\ p_{t,2} \end{pmatrix}.$$

Thereby $\tilde{w}$ is the vector of the smoothed inference regarding the chance of the Markov-Chain being in a particular state at a specific point in time. If now $\tilde{y}$ is regressed on $\tilde{x}$, with $\tilde{w}$ used as weights, then each column corresponds to one coefficient estimate, column one would correspond to $\beta_{1,1}$, the second column would correspond to $\beta_{1,2}$ and the third column would correspond to β_2. As becomes clear from this example, this algorithm does not follow the optimisation problems we propose in Example 1 to Example 5, to the best of the author's knowledge there is no proof of showing any equivalence. This fact brings us to our implementation of the EM algorithm in R, here we implement an approximation of the EM algorithm that strongly relies on our derivations in section 4.4.4 to section 4.4.8.

6 Building MSARM - Implementation Considerations

MSARM is the R package we developed utilizing the theoretical results presented in the previous sections, it can be installed with the command devtools::install_github("jmuelleo/MSARM"). MSARM

[2] Version 1.5, Sanchez-Espigares & Lopez-Moreno, 2021, DOI: 10.32614/CRAN.package.MSwM

[3] https://github.com/cran/MSwM/blob/master/R/2MSM.r

utilizes Example 3, Example 4, and Example 5 for estimating the parameters of any potential Markov-Switching AR process. One important point to note is that MSARM differs from the earlier presented theory in only one significant detail. Instead of solving a constraint optimisation problem, as the theory indicates, MSARM first computes the values of the constraint variables using the estimates from the previous iteration step and then computes the estimates of the parameters of the underlying process for the current iteration by inserting the constraint-variables into the originally constrained optimisation problem. Earlier attempts at developing MSARM included deriving the Gradient for the different setups and implementing gradient descent methods; this and any other form of constrained optimization led to higher computational demands without yielding better results. Therefore, we decided to utilize the described approximation. Besides this, the theory has been implemented in MSARM exactly as presented earlier. Summarizing we can say that MSARM allows its user to estimate Markov-Switching AR models. Any finite lag-order or number of underlying regimes can be chosen, details to the functions of MSARM can be found in the following code boxes and their descriptions.

6.1 MSARM.fit

MSARM.fit allows its user to estimate Markov-Switching AR models. Thereby the user only has to support MSARM.fit with the time series that is to be analyzed, the lag order of the assumed underlying AR process, as well as the number of assumed regimes and a "Switching"-vector, which indicates for all parameters (intercept, coefficients and error term variance), whether they are supposed to switch or not. It should be noted that if standard settings are used MSARM.fit estimates the parameters five times, each time starting the optimisation with a different random starting point. Utilizing set.seed before running MSARM.fit allows for full reproducibility of the results. The implemented performance metrics for choosing one of the optima are:

1. "LV": Utilizes the value of the conditional log-likelihood function for optima selection, i.e. the optimization attempt that maximized (23) is chosen.

2. "RSS": Utilizes the quality of the in-sample fit for optima selection, i.e. the optimization attempt that minimized $\sum_{t=K+1}^{T} \hat{u}_t^2$ is chosen, where K is the lag-order of the model and $\hat{u}_t$ are the residuals from the in-sample fit.

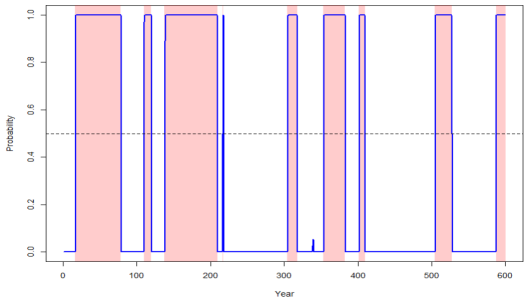
3. "RCM": Utilizes a Gini-Coefficient approach for optima selection, i.e. the optimization attempt that minimizes $100 \cdot N^2 \cdot \frac{1}{T-K} \sum_{t=K+1}^{T} P_{\hat{\lambda}(max)}(S_t = 1 | \mathscr{Y}_T = \vec{y}_T) \cdot \ldots \cdot P_{\hat{\lambda}(max)}(S_t = N | \mathscr{Y}_T = \vec{y}_T)$

4. "Entropy": Utilizes an Entropy approach for optima selection, i.e. the optimization attempt that minimizes $-100 \cdot N^2 \cdot \frac{1}{T-K} \sum_{t=K+1}^{T} P_{\hat{\lambda}(max)}(S_t = 1 | \mathscr{Y}_T = \vec{y}_T) \ln(P_{\hat{\lambda}(max)}(S_t = 1 | \mathscr{Y}_T = \vec{y}_T))$

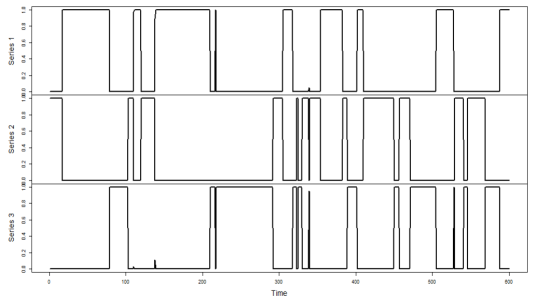
MSARM: MSARM.fit

```
MSARM.fit(Y_T = Y_T,              #Time Series to be analyzed
          K = K,                  #Lag Order
          N = N,                  #Number of Underlying Regimes
          m = 1,                  #Number of Observations to condition on
          Switcher = Switcher,    #Vector of TRUE/FALSE values of the length K+2
          threshold_value = 0.5, #Threshold for assigning regimes
          max_value = 250,        #Number of iterations
          R_value = 5,            #Number of random starting points
          Crit_value = "LV",      #Metric for choosing the optimisation result
          all.plot = FALSE)       #Set TRUE for plots of all optimisation results
```

MSARM.fit will give the user the following plots of the chosen optimisation results: First a plot of the regime probability of the second regime, this plot is particularily useful when working with two regimes and second a plot of the regime probability for all regimes, this plot is particularily useful when working with more than two regimes:



(a) MSARM.fit: Regime Probability Plot Type 1



(b) MSARM.fit: Regime Probability Plot Type 2

Figure 1: MSARM.fit: Regime Probability Plots

Additionally MSARM.fit will give a plot of the time series, the predicted regimes and the in-sample fit:

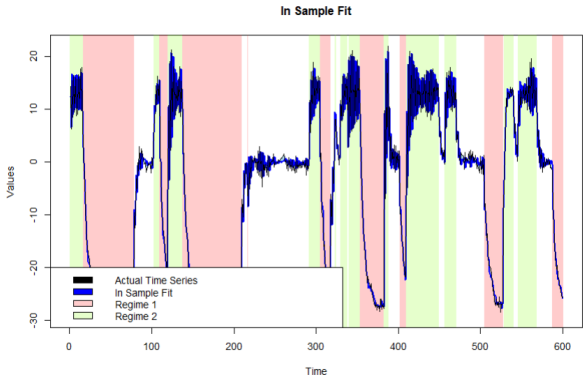


Figure 2: MSARM.fit: In-Sample Fit

6.2 MSARM.predict

MSARM.predict allows its user to predict an MSARM.fit output for n.ahead periods ahead. If boot is set to FALSE then the theory from section 4.5 and 4.6 will be implemented. Furthermore confidence intervals can be created by setting boot = TRUE, in that case L bootstrap estimates are calculated and used to create bootstrap-forecasting intervals. To be a bit more specific, bootstrapping is implemented in the following way:

Algorithm 1 Bootstrap Forecasts

Require: Time series data $y_T, ..., y_1$, number of repetitions L, number of forecast steps n.ahead and in-sample residuals $\hat{u}$

for $\ell \leftarrow 1$ to L **do**
  $y_{\text{data}} \leftarrow y_T, ..., y_1$ ▷ Start with original data
  for $h \leftarrow 1$ to n.ahead **do**
    $\hat{y}_{T+h} \leftarrow \text{Forecast}(y_{\text{data}})$
    $u \leftarrow \text{RandomSample}(\hat{u})$
    $y_{\text{new}} \leftarrow \hat{y}_{T+h} + u$
    Append y_{new} to y_{data}
  end for
  Store forecast path of y_{new} from $T+1$ to $T+n$.ahead
end for

After creating L forecast paths with this algorithm one can use the mean of the forecast paths as bootstrap forecast and the quantiles of the forecast paths as bootstrap confidence intervals.

MSARM: MSARM.predict

```
MSARM.predict(res_MSARM.fit,                    #Result from MSARM.fit
              n.ahead = 1,                      #Number of time periods
              boot = FALSE,                     #TRUE for bootstrapping
              levels = c(0.95,0.9,0.8,0.7,0.6), #Bootstrap interval levels
             ´ L = 10000)                       #Number of bootstrap estimates
```

6.3 MSARM.plot

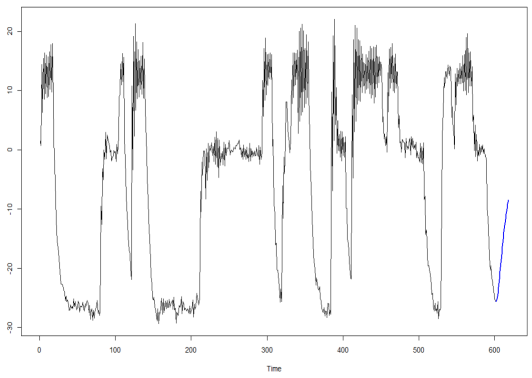
MSARM.plot allows its user to plot the forecasts from MSARM.predict to see how the time series is expected to behave in the future. Furthermore MSARM.plot allows its user if conf = TRUE and boot = TRUE (in MSARM.predict) to additionally plot the bootstrap confidence intervals for the forecasts.

MSARM: MSARM.plot

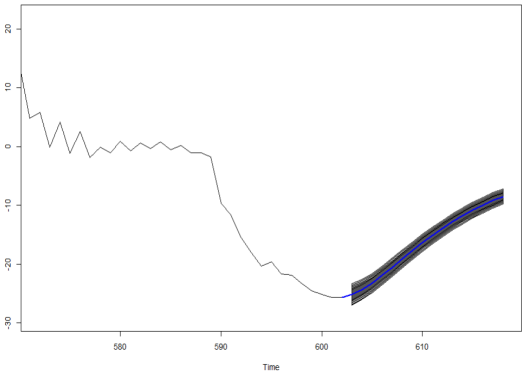
```
MSARM.plot(res_MSARM.predict, #Result from MSARM.predict
           conf = FALSE,      #TRUE for confidence intervals
           start = c(1,1),    #Beginning of the Time Series
           freq = 1)          #Number of seasons per time period
```

Standard forecast plots will include the observed time series and the forecasts in blue. If one chooses

to plot bootstrap confidence intervals, then each confidence level will be shown with a unique gray scale:



(a) MSARM.plot: Standard Forecasts



(b) MSARM.plot: Bootstrap Confidence Intervals

Figure 3: MSARM.plot

7 MSARM vs. MSwM

In the following, we present simulation results for the setups Example 0 to Example 5. It will become clear that MSwM struggles with certain setups, namely the more generalized setups where only a subset of the coefficients is allowed to switch. Additionally we present in the Appendix, section 9.4, around 300 more simulation results, where we compared the performance of MSARM and MSwM utilizing completely randomly generated processes. But first, we turn to some explicit examples of applying MSARM and MSwM.

7.1 Example 0: Switching Coefficients and Intercept, Non-Switching σ^2

We simulated 300 observations of the following process:

$$Y_t = c_{s_t} + \phi_{1,s_t} Y_{t-1} + \phi_{2,s_t} Y_{t-2} + U_t; \quad \text{where} \quad U_t \overset{i.i.d.}{\sim} N(0, \sigma^2).$$

Furthermore, the parameter vector α together with the transition matrix Π have the following form:

	c_{s_t}	ϕ_{1,s_t}	ϕ_{2,s_t}	σ^2	π_1	π_2
Regime 1	-0.6	-0.3	0.3	1	0.95	0.05
Regime 2	0.6	0.3	-0.3	1	0.05	0.95

Table 1: Example 0: Parameter Values

The following graphics show the simulated process, as well as the the predicted regime probabilites by MSARM with standard settings and the predicted regime probabilites by MSwM.

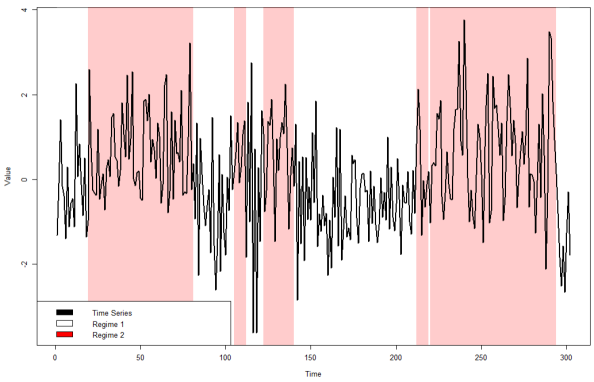
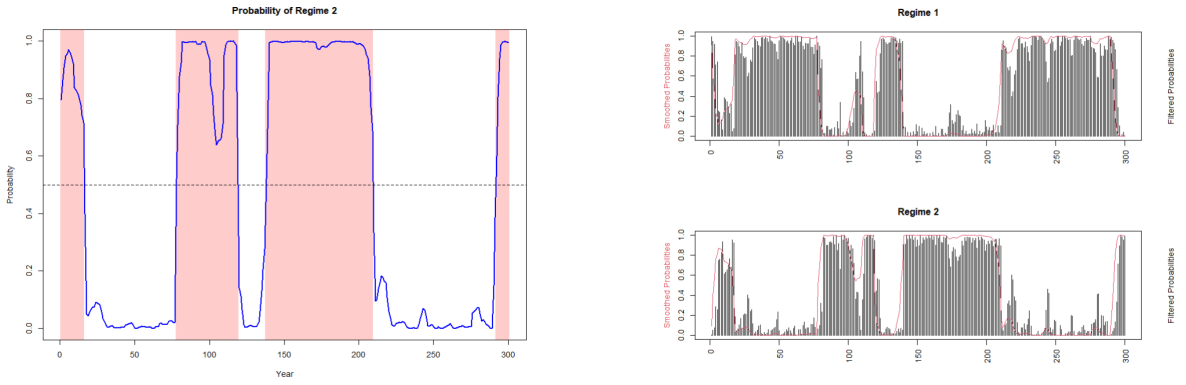


Figure 4: Example 0: Simulation



(a) Example 0: MSARM Regime Probability

(b) Example 0: MSwM Regime Probability

Figure 5: Example 0: Regime Probability MSARM vs MSwM

As one can easily see both packages lead to very similar results for Example 0, MSARM leads to a slightly lower misclassification rate for the regimes, while MSwM leads to slightly better estimates of the parameters, but the difference is very small. The exact resulting estimates and performance indicators can be found in the following tables, the performance indicators computed were the misclassification rate (MCR), the mean absolute coefficient estimation error (ACoEE), the mean absolute transition matrix estimation error (APiEE), the mean absolute error term variance estimation error (AVarEE), and the mean absolute parameter estimation error (APaEE).

	c_{s_t}	ϕ_{1,s_t}	ϕ_{2,s_t}	σ^2	π_1	π_2
MSARM Regime 1	−0.4370	−0.1712	0.3706	1.0890	0.9690	0.0310
MSARM Regime 2	0.6030	0.2816	−0.2524	1.0890	0.0283	0.9717
MSwM Regime 1	−0.4503	−0.1761	0.3656	1.0862	0.9595	0.0405
MSwM Regime 2	0.5904	0.2779	−0.2459	1.0862	0.0261	0.9739

Table 2: Example 0: Estimated Parameter Values

	MCR	ACoEE	APiEE	AVarEE	APaEE
MSARM	0.0300	0.0719	0.0203	0.0890	0.0576
MSwM	0.0400	0.0709	0.0167	0.0862	0.0554

Table 3: Example 0: Performance Metrics

7.2 Example 1: Non-Switching Intercept

We simulated 300 observations of the following process:

$$Y_t = c + \phi_{1,s_t} Y_{t-1} + \phi_{2,s_t} Y_{t-2} + U_t \quad \text{where} \quad U_t \overset{i.i.d.}{\sim} N(0, \sigma^2).$$

Furthermore the parameter vector α together with the transition matrix Π have the following form:

	c	ϕ_{1,s_t}	ϕ_{2,s_t}	σ^2	π_1	π_2
Regime 1	0.3	−0.4	0.4	1	0.95	0.05
Regime 2	0.3	0.5	−0.5	1	0.05	0.95

Table 4: Example 1: Parameter Values

The following graphics show the simulated process, as well as the the predicted regime probabilites by MSARM with standard settings and the predicted regime probabilites by MSwM.

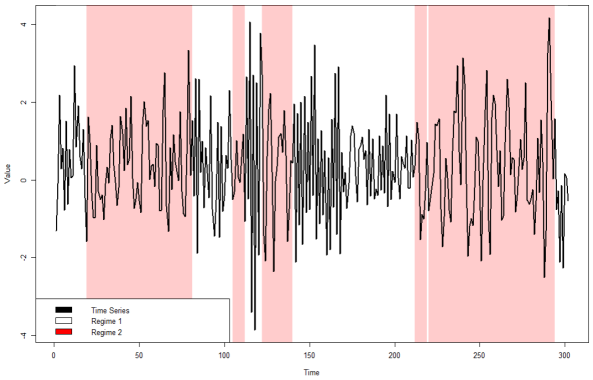
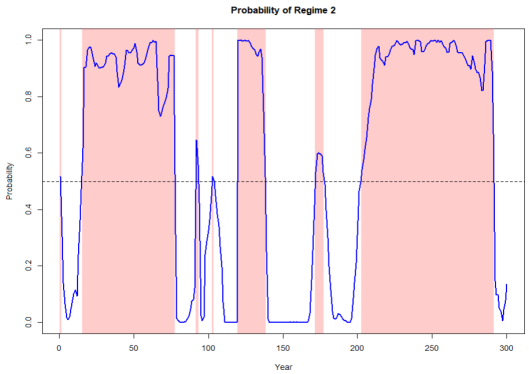
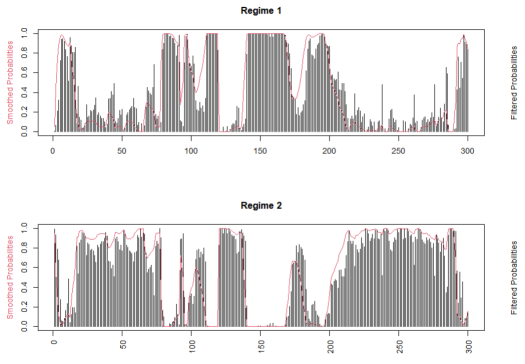


Figure 6: Example 1: Simulation



(a) Example 1: MSARM Regime Probability



(b) Example 1: MSwM Regime Probability

Figure 7: Example 1: Regime Probability MSARM vs MSwM

As one can easily see, both packages lead to similar results for Example 1. MSARM results in a slightly lower misclassification rate for the regimes and slightly better estimates of the parameters; however, the difference is again quite small. The exact resulting estimates and performance indicators can be found in the following tables:

	c	ϕ_{1,s_t}	ϕ_{2,s_t}	σ^2	π_1	π_2
MSARM Regime 1	0.3118	−0.3657	0.4596	1.0281	0.9408	0.0592
MSARM Regime 2	0.3118	0.4691	−0.4610	1.0281	0.0417	0.9583
MSwM Regime 1	0.3181	−0.3651	0.4612	1.0242	0.9280	0.0720
MSwM Regime 2	0.3182	0.4622	−0.4637	1.0242	0.0451	0.9549

Table 5: Example 1: Estimated Parameter Values

	MCR	ACoEE	APiEE	AVarEE	APaEE
MSARM	0.0867	0.0312	0.0088	0.0281	0.0232
MSwM	0.0933	0.0344	0.0135	0.0242	0.0257

Table 6: Example 1: Performance Metrics

7.3 Example 2: Switching Intercept and Non-Switching Coefficients

We simulated 300 observations of the following process:

$$y_t = c_{s_t} + \phi_1 y_{t-1} + \phi_2 y_{t-2} + U_t; \quad \text{where} \quad U_t \overset{i.i.d.}{\sim} N(0, \sigma^2).$$

Furthermore, the parameter vector α together with the transition matrix Π have the following form:

	c_{s_t}	ϕ_1	ϕ_2	σ^2	π_1	π_2
Regime 1	2	−0.4	0.5	1	0.95	0.05
Regime 2	−2	−0.4	0.5	1	0.05	0.95

Table 7: Example 2: Parameter Values

The following graphics show the simulated process, as well as the predicted regime probabilites by MSARM with standard settings and the predicted regime probabilites by MSwM.

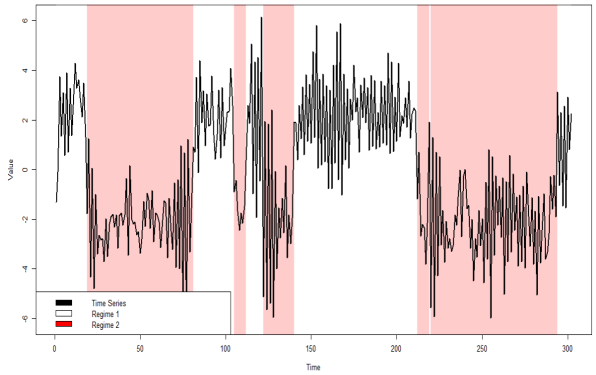
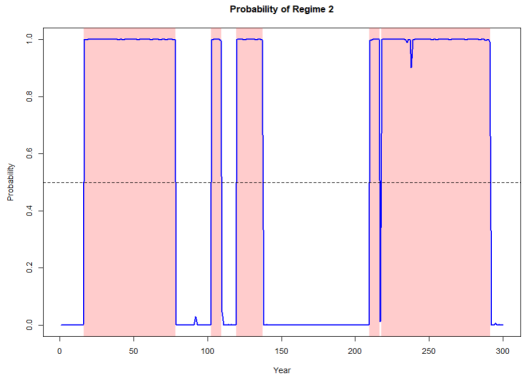
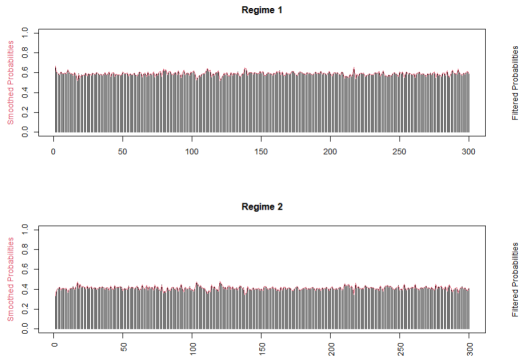


Figure 8: Example 2: Simulation



(a) Example 2: MSARM Regime Probability



(b) Example 2: MSwM Regime Probability

Figure 9: Example 2: Regime Probability MSARM vs MSwM

This is now the first explicit example where MSwM completely breaks down and fails to accurately estimate the underlying process. It is essential to emphasize that MSwM does not just fail in this specific example; instead, it tends to fail in setups of the Example 2 type in general, as further illustrated in the Appendix, section 9.4. Unfortunately, Example 2 is the setup most similar to the one used by Hamilton for estimating recession probabilities and is therefore particularly interesting, see Hamilton (1994, page 697). Furthermore, we want to emphasize that MSARM does particualarily well in setups like this, even reaching a misclassifcation rate of exactly 0%. The exact resulting estimates and performance indicators can be found in the following tables:

	c_{s_t}	ϕ_1	ϕ_2	σ^2	π_1	π_2
MSARM Regime 1	2.0684	−0.3789	0.4558	1.0371	0.9607	0.0393
MSARM Regime 2	−2.0165	−0.3789	0.4558	1.0371	0.0306	0.9694
MSwM Regime 1	0.0341	0.0532	0.8272	2.1143	0.6111	0.3889
MSwM Regime 2	−0.0850	0.0532	0.8272	2.1143	0.5579	0.4421

Table 8: Example 2: Estimated Parameter Values

	MCR	ACoEE	APiEE	AVarEE	APaEE
MSARM	0	0.0359	0.0150	0.0371	0.0291
MSwM	0.4400	0.9069	0.4234	1.1143	0.7803

Table 9: Example 2: Performance Metrics

7.4 Example 3: All Parameters Switch

We simulated 300 observations of the following process:

$$Y_t = c_{s_t} + \phi_{1,s_t} Y_{t-1} + \phi_{2,s_t} Y_{t-2} + U_t; \quad \text{where} \quad U_t \sim N(0, \sigma^2_{s_t}).$$

Furthermore, the parameter vector α together with the transition matrix Π have the following form:

	c_{s_t}	ϕ_{1,s_t}	ϕ_{2,s_t}	$\sigma^2_{s_t}$	π_1	π_2
Regime 1	2	-0.4	-0.5	1	0.95	0.05
Regime 2	-2	0.4	0.5	9	0.05	0.95

Table 10: Example 3: Parameter Values

The following graphics show the simulated process, as well as the predicted regime probabilites by MSARM with standard settings and the predicted regime probabilites by MSwM.

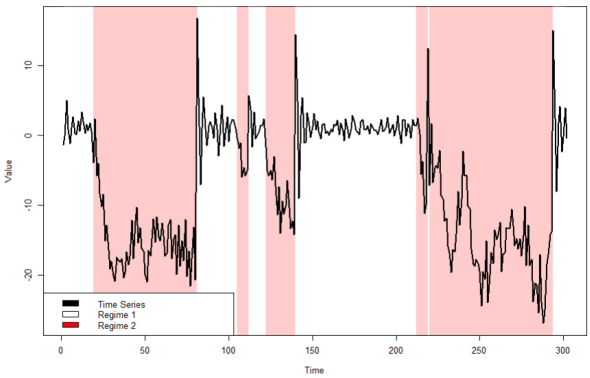
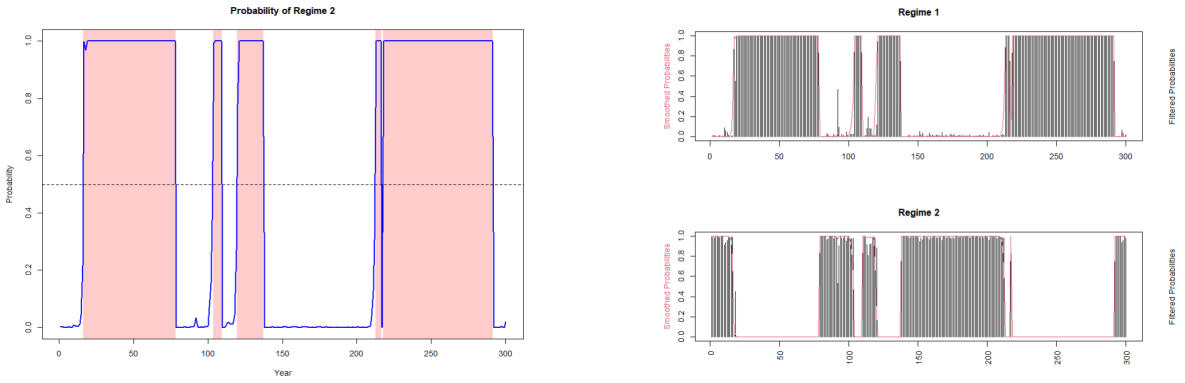


Figure 10: Example 3: Simulation



(a) Example 3: MSARM Regime Probability

(b) Example 3: MSwM Regime Probability

Figure 11: Example 3: Regime Probability MSARM vs MSwM

In this setup MSARM slightly outperforms MSwM in every metric, besides MCR where both achieve the same result, but still both packages perform relatively similar. The exact resulting estimates and performance indicators can be found in the following tables:

	c_{s_t}	ϕ_{1,s_t}	ϕ_{2,s_t}	$\sigma^2_{s_t}$	π_1	π_2
MSARM Regime 1	2.0271	-0.3785	-0.5230	0.9918	0.9613	0.0387
MSARM Regime 2	-3.3138	0.3356	0.4577	9.3624	0.0311	0.9689
MSwM Regime 1	2.0506	-0.3780	-0.5245	1.0212	0.9618	0.0382
MSwM Regime 2	-3.3198	0.3354	0.4576	9.3608	0.0311	0.9689

Table 11: Example 3: Esimated Parameter Values

	MCR	ACoEE	APiEE	AVarEE	APaEE
MSARM	0.0133	0.2487	0.0151	0.1853	0.1603
MSwM	0.0133	0.2540	0.0153	0.1910	0.1639

Table 12: Example 3: Performance Metrics

7.5 Example 4: Arbitrary Subset Switching of (c, ϕ) and Non-Switching σ^2

We simulated 300 observations of the following process:

$$Y_t = c_{s_t} + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \phi_3 Y_{t-3} + \phi_{4,s_t} Y_{t-4} + U_t; \quad \text{where} \quad U_t \overset{i.i.d.}{\sim} N(0, \sigma^2).$$

Furthermore the parameter vector α together with the transition matrix Π have the following form:

	c_{s_t}	ϕ_1	ϕ_2	ϕ_3	ϕ_{4,s_t}	σ^2	π_1	π_2
Regime 1	3	-0.3	0.3	0.2	-0.6	1	0.95	0.05
Regime 2	-3	-0.3	0.3	0.2	0.6	1	0.05	0.95

Table 13: Example 4: Parameter Values

The following graphics show the simulated process, as well as the the predicted regime probabilites by MSARM with standard settings and the predicted regime probabilites by MSwM.

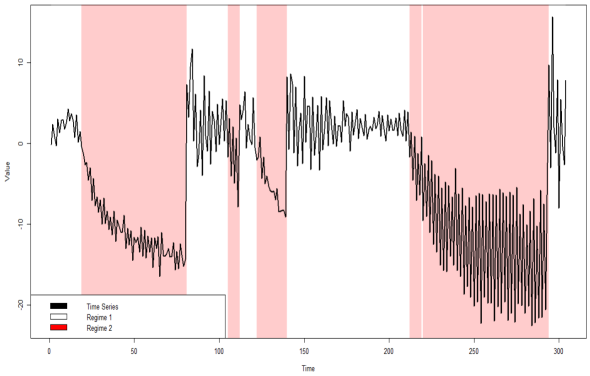
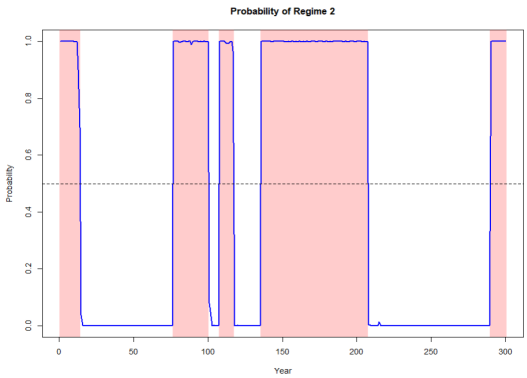
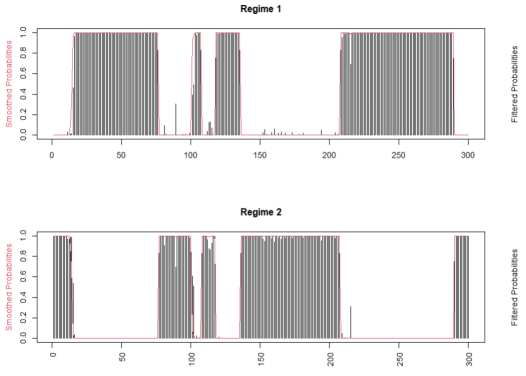


Figure 12: Example 4: Simulation



(a) Example 4: MSARM Regime Probability



(b) Example 4: MSwM Regime Probability

Figure 13: Example 4: Regime Probability MSARM vs MSwM

For this setup MSARM performs as well as MSwM regarding the misclassification rate, but performs slightly worse regarding the parameter estimation, but still both packages perform very similar. The exact resulting estimates and performance indicators can be found in the following tables:

	c_{s_t}	ϕ_1	ϕ_2	ϕ_3	ϕ_{4,s_t}	σ^2	π_1	π_2
MSARM Regime 1	3.0273	−0.3026	0.2648	0.1882	−0.5575	1.0751	0.9687	0.0313
MSARM Regime 2	−3.1302	−0.3026	0.2648	0.1882	0.6301	1.0751	0.0239	0.9761
MSwM Regime 1	3.0232	−0.3016	0.2647	0.1874	−0.5568	1.0606	0.9690	0.0310
MSwM Regime 2	−3.1288	−0.3016	0.2647	0.1874	0.6302	1.0606	0.0239	0.9761

Table 14: Example 4: Estimated Parameter Values

	MCR	ACoEE	APiEE	AVarEE	APaEE
MSARM	0.0033	0.0329	0.0224	0.0751	0.0356
MSwM	0.0033	0.0325	0.0225	0.0606	0.0335

Table 15: Example 4: Performance Metrics

7.6 Example 5: Arbitrary Subset-Switching of (c, ϕ) and Switching σ^2

We simulated 300 observations of the following process:

$$y_t = c_{s_t} + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + U_t; \quad \text{where} \quad U_t \sim N(0, \sigma^2_{s_t}).$$

Furthermore the parameter vector α together with the transition matrix Π have the following form:

	c_{s_t}	ϕ_1	ϕ_2	$\sigma^2_{s_t}$	π_1	π_2
Regime 1	7	−0.6	0.4	1	0.95	0.05
Regime 2	−7	−0.6	0.4	4	0.05	0.95

Table 16: Example 5: Parameter Values

The following graphics show the simulated process, as well as the the predicted regime probabilites by MSARM with standard settings and the predicted regime probabilites by MSwM.

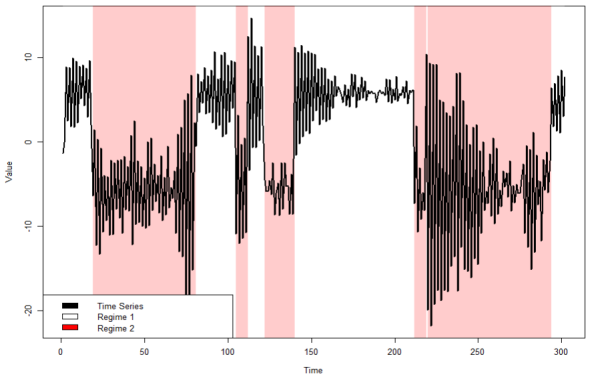
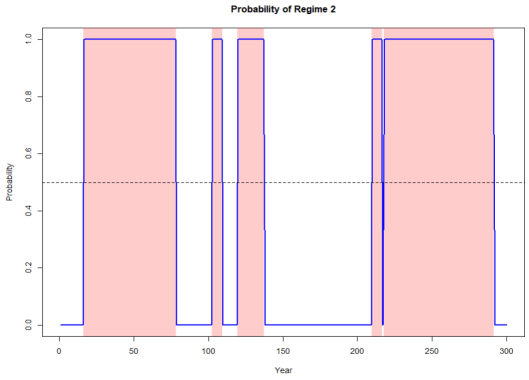
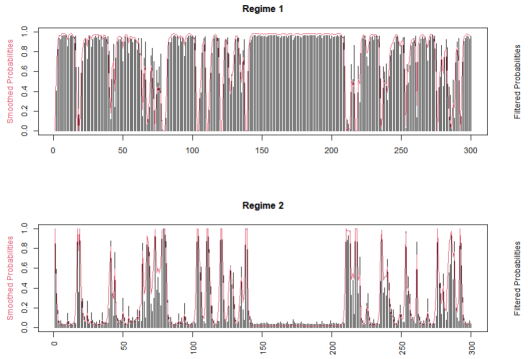


Figure 14: Example 5: Simulation



(a) Example 5: MSARM Regime Probability



(b) Example 5: MSwM Regime Probability

Figure 15: Example 5: Regime Probability MSARM vs MSwM

This is the second explicit example where MSwM fails to estimate the underlying process properly while MSARM does so without difficulties. In contrast MSARM does particularily well achieving a misclassification rate of exactly 0%. The exact resulting estimates and performance indicators can be found in the following tables:

	c_{s_t}	ϕ_1	ϕ_2	$\sigma^2_{s_t}$	π_1	π_2
MSARM Regime 1	7.0555	−0.5688	0.3610	0.9158	0.9618	0.0382
MSARM Regime 2	−6.9812	−0.5688	0.3610	4.2233	0.0298	0.9702
MSwM Regime 1	0.1615	−0.0005	0.8963	2.5668	0.8936	0.1064
MSwM Regime 2	−0.6526	−0.0005	0.8963	39.7116	0.3270	0.6730

Table 17: Example 5: Estimated Parameter Values

	MCR	ACoEE	APiEE	AVarEE	APaEE
MSARM	0	0.0358	0.0160	0.1538	0.0489
MSwM	0.4533	2.5629	0.1667	18.6392	4.4436

Table 18: Example 5: Performance Metrics

8 Conclusion

In this thesis, we first provided a comprehensive and structured overview of the theoretical foundations for estimating Markov-Switching Autoregressive (AR) models. Building on this framework, we introduced MSARM, an R package developed as part of this Bachelor thesis for estimating such models within the R environment. As discussed earlier, MSARM heavily relies on the presented theory, particularly the generalizations covered in Examples 3, 4, and 5. Subsequently, we compared MSARM with MSwM, one of the most widely used R packages for Markov-Switching models, using approximately 300 simulation runs based on randomly generated processes. The results show that MSARM performs comparably or better where both packages succeed and significantly outperforms MSwM in terms of robustness. To be more specific, MSwM failed in 70 cases to estimate the underlying process sufficiently well, where MSARM succeeded, while MSARM failed in only 7 cases where MSwM

succeeded, implying a 10:1 advantage for MSARM. These results were achieved using MSARM's standard settings. As discussed in section 6.1, further performance improvements are possible by increasing the number of starting points or adjusting performance metrics. Overall, MSARM proves to be a robust and practical alternative to MSwM, especially when reducing the risk of estimation failure is critical.

References

Hamilton, J.D. (1989). *A New Approach to the Economic Analysis of Nonstationary Time Series and the Business Cycle*. Econometrica, **Volume 57**, pages 357-384.

Hamilton, J.D. (1990): *Analysis of Time Series subject to Changes in Regime*. Journal of Econometrics, **Volume 45**, pages 39-70.

Hamilton, J.D. (1994): *Time Series Analysis*. Princeton, NJ: Princeton University Press.

Kim, C.-J. (1994): *Dynamic linear models with Markov-switching*. Journal of Econometrics, **Volume 60**, pages 1-22.

Lee, S.-H. (2022). *Understanding Hamilton Regime Switching Model using R package*. R-bloggers: https://www.r-bloggers.com/2022/02/understanding-hamilton-regime-switching-model-using-r-package/ (Accessed: 01.07.2025).

Sanchez-Espigares, J.A., & Lopez-Moreno, A. (2021): *MSwM: Fitting Markov Switching Models* [R package source code]. GitHub: https://github.com/cran/MSwM/blob/master/R/2MSM.r (Accessed: 01.07.2025).

9 Appendix

9.1 Optimal Inference of the Regimes and Derivation of the Log-Likelihood

The following derivation closely follows Hamilton (1994, page 693). First of all, it is essential to keep in mind that $(\hat{\xi}_{t|t-1})_j = P_\theta(S_t = j|\mathscr{Y}_{t-1} = \vec{y}_{t-1})$ and $(\eta_t)_j = f_{Y_t|S_t,\mathscr{Y}_{t-1};\alpha}(y_t|j,\vec{y}_{t-1})$. Based on this we can see that:

$$\begin{aligned}(\hat{\xi}_{t|t-1} \odot \eta_t)_j &= P_\theta(S_t = j|\mathscr{Y}_{t-1} = \vec{y}_{t-1}) f_{Y_t|S_t,\mathscr{Y}_{t-1};\alpha}(y_t|j,\vec{y}_{t-1}) \\ &= f_{Y_t,S_t|\mathscr{Y}_{t-1};\theta}(y_t, j|\vec{y}_{t-1}).\end{aligned}$$

If we now sum over all potential values of S_t we get:

$$\sum_{j=1}^{N} f_{Y_t,S_t|\mathscr{Y}_{t-1};\theta}(y_t, j|\vec{y}_{t-1}) = f_{Y_t|\mathscr{Y}_{t-1};\theta}(y_t|\vec{y}_{t-1}) = \mathbf{1}'(\hat{\xi}_{t|t-1} \odot \eta_t).$$

Additionally it is therefore true that:

$$\frac{(\hat{\xi}_{t|t-1} \odot \eta_t)_j}{\mathbf{1}'(\hat{\xi}_{t|t-1} \odot \eta_t)} = \frac{f_{Y_t,S_t|\mathscr{Y}_{t-1};\theta}(y_t, j|\vec{y}_{t-1})}{f_{Y_t|\mathscr{Y}_{t-1};\theta}(y_t|\vec{y}_{t-1})} = P_\theta(S_t = j|\mathscr{Y}_t = \vec{y}_t) = (\hat{\xi}_{t|t})_j.$$

Utilizing vectors we can write:

$$\hat{\xi}_{t|t} = \frac{(\hat{\xi}_{t|t-1} \odot \eta_t)}{\mathbf{1}'(\hat{\xi}_{t|t-1} \odot \eta_t)}.$$

To put it as simple as possible one could say that we basically just applied Bayes Rule. Next we want to get from $(\hat{\xi}_{t|t})_j = P_\theta(S_t = j|\mathscr{Y}_t = \vec{y}_t)$ to $(\hat{\xi}_{t+1|t})_j = P_\theta(S_{t+1} = j|\mathscr{Y}_t = \vec{y}_t)$. We know from (14) that this is possible via:

$$E_\theta(\zeta_{t+1}|\mathscr{Y}_t = \vec{y}_t) = \Pi'\hat{\xi}_{t|t}.$$

9.2 Smoothed Inference over the Regimes

The following derivation closely follows Hamilton (1994, page 700-702). First we note that S_t depends on $\mathscr{Y}_{t-1}$ **only** through S_{t-1} and on future observations only through S_{t+1}! One could say:

$$P_\theta(S_t = j|S_{t+1} = i, \mathscr{Y}_T = \vec{y}_T) = P_\theta(S_t = j|S_{t+1} = i, \mathscr{Y}_t = \vec{y}_t).$$

We can show this formally:

$$\begin{aligned}
P_\theta(S_t = j|S_{t+1} = i, \mathscr{Y}_{t+1} = \vec{y}_{t+1}) &= P_\theta(S_t = j|S_{t+1} = i, Y_{t+1} = y_{t+1}, \mathscr{Y}_t = \vec{y}_t) \\
&= \frac{P_\theta(S_t = j, Y_{t+1} = y_{t+1}|S_{t+1} = i, \mathscr{Y}_t = \vec{y}_t)}{f_{Y_{t+1}|S_{t+1},\mathscr{Y}_t;\alpha}(y_{t+1}|i,\vec{y}_t)} \\
&= \frac{f_{Y_{t+1}|S_t,S_{t+1},\mathscr{Y}_t;\alpha}(y_{t+1}|j,i,\vec{y}_t)}{f_{Y_{t+1}|S_{t+1},\mathscr{Y}_t;\alpha}(y_{t+1}|i,\vec{y}_t)} P_\theta(S_t = j|S_{t+1} = i, \mathscr{Y}_t = \vec{y}_t) \\
&= \frac{f_{Y_{t+1}|S_{t+1},\mathscr{Y}_t;\alpha}(y_{t+1}|i,\vec{y}_t)}{f_{Y_{t+1}|S_{t+1},\mathscr{Y}_t;\alpha}(y_{t+1}|i,\vec{y}_t)} P_\theta(S_t = j|S_{t+1} = i, \mathscr{Y}_t = \vec{y}_t) \\
&= P_\theta(S_t = j|S_{t+1} = i, \mathscr{Y}_t = \vec{y}_t).
\end{aligned}$$

The last two steps are possible because based on (11) the distribution of Y_{t+1} conditional on S_{t+1} is independent of S_t. Now we can approach the derivation for $t+2$ in a similar way:

$$\begin{aligned}
P_\theta(S_t = j|S_{t+1} = i, \mathscr{Y}_{t+2} = \vec{y}_{t+2}) &= P_\theta(S_t = j|S_{t+1} = i, Y_{t+2} = y_{t+2}, \mathscr{Y}_{t+1} = \vec{y}_{t+1}) \\
&= \frac{P_\theta(S_t = j, Y_{t+2} = y_{t+2}|S_{t+1} = i, \mathscr{Y}_{t+1} = \vec{y}_{t+1})}{f_{Y_{t+2}|S_{t+1},\mathscr{Y}_{t+1};\theta}(y_{t+2}|i,\vec{y}_{t+1})} \\
&= \frac{f_{Y_{t+2}|S_t,S_{t+1},\mathscr{Y}_{t+1};\theta}(y_{t+2}|j,i,\vec{y}_{t+1})}{f_{Y_{t+2}|S_{t+1},\mathscr{Y}_{t+1};\theta}(y_{t+2}|i,\vec{y}_{t+1})} P_\theta(S_t = j|S_{t+1} = i, \mathscr{Y}_{t+1} = \vec{y}_{t+1}) \\
&= \frac{f_{Y_{t+2}|S_{t+1},\mathscr{Y}_{t+1};\theta}(y_{t+2}|i,\vec{y}_{t+1})}{f_{Y_{t+2}|S_{t+1},\mathscr{Y}_{t+1};\theta}(y_{t+2}|i,\vec{y}_{t+1})} P_\theta(S_t = j|S_{t+1} = i, \mathscr{Y}_{t+1} = \vec{y}_{t+1}) \\
&= P_\theta(S_t = j|S_{t+1} = i, \mathscr{Y}_{t+1} = \vec{y}_{t+1}) \\
&= P_\theta(S_t = j|S_{t+1} = i, \mathscr{Y}_t = \vec{y}_t).
\end{aligned}$$

Simplifying the fraction is possible because:

$$\begin{aligned}
f_{Y_{t+2}|S_t,S_{t+1},\mathscr{Y}_{t+1};\theta}(y_{t+2}|j,i,\vec{y}_{t+1}) &= \sum_{k=1}^{N} f_{Y_{t+2},S_{t+2}|S_t,S_{t+1},\mathscr{Y}_{t+1};\theta}(y_{t+2},k|j,i,\vec{y}_{t+1}) \\
&= \sum_{k=1}^{N} f_{Y_{t+2}|S_{t+2},S_t,S_{t+1},\mathscr{Y}_{t+1};\alpha}(y_{t+2}|k,j,i,\vec{y}_{t+1}) \\
&\quad \cdot P_\theta(S_{t+2}=k|S_{t+1}=i,S_t=j,\mathscr{Y}_{t+1}=\vec{y}_{t+1}) \\
&= \sum_{k=1}^{N} f_{Y_{t+2}|S_{t+2},S_{t+1},\mathscr{Y}_{t+1};\alpha}(y_{t+2}|k,i,\vec{y}_{t+1}) \\
&\quad \cdot P_\theta(S_{t+2}=k|S_{t+1}=i,\mathscr{Y}_{t+1}=\vec{y}_{t+1}) \\
&= \sum_{k=1}^{N} f_{Y_{t+2},S_{t+2}|S_{t+1},\mathscr{Y}_{t+1};\theta}(y_{t+2},k|i,\vec{y}_{t+1}) \\
&= f_{Y_{t+2}|S_{t+1},\mathscr{Y}_{t+1};\theta}(y_{t+2}|i,\vec{y}_{t+1}).
\end{aligned}$$

We show now by induction that this approach is generally applicable. The earlier shown cases were the start of the induction, for the induction step we can say, we choose an arbitrary, but feasible, n and our induction assumption is:

$$P_\theta(S_t=j|S_{t+1}=i,\mathscr{Y}_{t+n}=\vec{y}_{t+n}) = P_\theta(S_t=j|S_{t+1}=i,\mathscr{Y}_t=\vec{y}_t). \tag{53}$$

Then it shall hold that:

$$P_\theta(S_t=j|S_{t+1}=i,\mathscr{Y}_{t+n+1}=\vec{y}_{t+n+1}) = P_\theta(S_t=j|S_{t+1}=i,\mathscr{Y}_t=\vec{y}_t). \tag{54}$$

To show that we write:

$$\begin{aligned}
P_\theta(S_t=j|S_{t+1}=i,\mathscr{Y}_{t+n+1}=\vec{y}_{t+n+1}) &= P_\theta(S_t=j|S_{t+1}=i,Y_{t+n+1}=y_{t+n+1},\mathscr{Y}_{t+n}=\vec{y}_{t+n}) \\
&= \frac{f_{S_t,Y_{t+n+1}|S_{t+1},\mathscr{Y}_{t+n};\theta}(j,y_{t+n+1}|i,\vec{y}_{t+n})}{f_{Y_{t+n+1}|S_{t+1},\mathscr{Y}_{t+n};\theta}(y_{t+n+1}|i,\vec{y}_{t+n})} \\
&= \frac{f_{Y_{t+n+1}|S_t,S_{t+1},\mathscr{Y}_{t+n};\theta}(y_{t+n+1}|j,i,\vec{y}_{t+n})}{f_{Y_{t+n+1}|S_{t+1},\mathscr{Y}_{t+n};\theta}(y_{t+n+1}|i,\vec{y}_{t+n})} \\
&\quad \cdot P_\theta(S_t=j|S_{t+1}=i,\mathscr{Y}_{t+n}=\vec{y}_{t+n}) \\
&= P_\theta(S_t=j|S_{t+1}=i,\mathscr{Y}_{t+n}=\vec{y}_{t+n}) \\
&= P_\theta(S_t=j|S_{t+1}=i,\mathscr{Y}_t=\vec{y}_t).
\end{aligned}$$

Thereby, the last step is just applying the induction assumption and simplifying the fraction is possible

because:

$$\begin{aligned}
f_{Y_{t+n+1}|S_t,S_{t+1},\mathscr{Y}_{t+n};\theta}(y_{t+n+1}|j,i,\vec{y}_{t+n}) &= \sum_{k=1}^{N} f_{Y_{t+n+1},S_{t+n+1}|S_t,S_{t+1},\mathscr{Y}_{t+n};\theta}(y_{t+n+1},k|j,i,\vec{y}_{t+n}) \\
&= \sum_{k=1}^{N} f_{Y_{t+n+1}|S_{t+n+1},S_t,S_{t+1},\mathscr{Y}_{t+n};\alpha}(y_{t+n+1}|k,j,i,\vec{y}_{t+n}) \\
&\quad \cdot P_\theta(S_{t+n+1}=k|S_t=j,S_{t+1}=i,\mathscr{Y}_{t+n}=\vec{y}_{t+n}) \\
&= \sum_{k=1}^{N} f_{Y_{t+n+1}|S_{t+n+1},S_{t+1},\mathscr{Y}_{t+n};\alpha}(y_{t+n+1}|k,i,\vec{y}_{t+n}) \\
&\quad \cdot P_\theta(S_{t+n+1}=k|S_{t+1}=i,\mathscr{Y}_{t+n}=\vec{y}_{t+n}) \\
&= \sum_{k=1}^{N} f_{Y_{t+n+1},S_{t+n+1}|S_{t+1},\mathscr{Y}_{t+n};\theta}(y_{t+n+1},k|i,\vec{y}_{t+n}) \\
&= f_{Y_{t+n+1}|S_{t+1},\mathscr{Y}_{t+n};\theta}(y_{t+n+1}|i,\vec{y}_{t+n}).
\end{aligned}$$

Once this is done it can be seen that:

$$P_\theta(S_t=j|S_{t+1}=i,\mathscr{Y}_{t+m}=\vec{y}_{t+m}) = P_\theta(S_t=j|S_{t+1}=i,\mathscr{Y}_t=\vec{y}_t).$$

From this follows the original claim:

$$P_\theta(S_t=j|S_{t+1}=i,\mathscr{Y}_T=\vec{y}_T) = P_\theta(S_t=j|S_{t+1}=i,\mathscr{Y}_t=\vec{y}_t).$$

Next we can see that:

$$\begin{aligned}
P_\theta(S_t=j|S_{t+1}=i,\mathscr{Y}_t=\vec{y}_t) &= \frac{P_\theta(S_t=j,S_{t+1}=i|\mathscr{Y}_t=\vec{y}_t)}{P_\theta(S_{t+1}=i|\mathscr{Y}_t=\vec{y}_t)} \\
&= \frac{P_\theta(S_{t+1}=i|S_t=j,\mathscr{Y}_t=\vec{y}_t)P_\theta(S_t=j|\mathscr{Y}_t=\vec{y}_t)}{P_\theta(S_{t+1}=i|\mathscr{Y}_t=\vec{y}_t)} \\
&= \frac{P_\theta(S_{t+1}=i|S_t=j)P_\theta(S_t=j|\mathscr{Y}_t=\vec{y}_t)}{P_\theta(S_{t+1}=i|\mathscr{Y}_t=\vec{y}_t)} \\
&= \frac{\pi_{j,i}P_\theta(S_t=j|\mathscr{Y}_t=\vec{y}_t)}{P_\theta(S_{t+1}=i|\mathscr{Y}_t=\vec{y}_t)}.
\end{aligned}$$

From this it follows that:

$$\begin{aligned}
P_\theta(S_t=j,S_{t+1}=i|\mathscr{Y}_T=\vec{y}_T) &= P_\theta(S_{t+1}=i|\mathscr{Y}_T=\vec{y}_T)P_\theta(S_t=j|S_{t+1}=i,\mathscr{Y}_T=\vec{y}_T) \\
&= P_\theta(S_{t+1}=i|\mathscr{Y}_T=\vec{y}_T)P_\theta(S_t=j|S_{t+1}=i,\mathscr{Y}_t=\vec{y}_t) \\
&= P_\theta(S_{t+1}=i|\mathscr{Y}_T=\vec{y}_T)\frac{\pi_{j,i}P_\theta(S_t=j|\mathscr{Y}_t=\vec{y}_t)}{P_\theta(S_{t+1}=i|\mathscr{Y}_t=\vec{y}_t)}.
\end{aligned}$$

Therefore, the smoothed inference over S_t is given by:

$$\begin{aligned}
P_\theta(S_t = j|\mathscr{Y}_T = \vec{y}_T) &= \sum_{i=1}^{N} P_\theta(S_t = j, S_{t+1} = i|\mathscr{Y}_T = \vec{y}_T) \\
&= \sum_{i=1}^{N} P_\theta(S_{t+1} = i|\mathscr{Y}_T = \vec{y}_T) \frac{\pi_{j,i} P_\theta(S_t = j|\mathscr{Y}_t = \vec{y}_t)}{P_\theta(S_{t+1} = i|\mathscr{Y}_t = \vec{y}_t)} \\
&= P_\theta(S_t = j|\mathscr{Y}_t = \vec{y}_t) \sum_{i=1}^{N} \frac{\pi_{j,i} P_\theta(S_{t+1} = i|\mathscr{Y}_T = \vec{y}_T)}{P_\theta(S_{t+1} = i|\mathscr{Y}_t = \vec{y}_t)} \\
&= P_\theta(S_t = j|\mathscr{Y}_t = \vec{y}_t)(\pi_{j,1} \ldots \pi_{j,N}) \begin{pmatrix} \frac{P_\theta(S_{t+1} = 1|\mathscr{Y}_T = \vec{y}_T)}{P_\theta(S_{t+1} = 1|\mathscr{Y}_t = \vec{y}_t)} \\ \ldots \\ \frac{P_\theta(S_{t+1} = N|\mathscr{Y}_T = \vec{y}_T)}{P_\theta(S_{t+1} = N|\mathscr{Y}_t = \vec{y}_t)} \end{pmatrix} \\
&= P_\theta(S_t = j|\mathscr{Y}_t = \vec{y}_t) \Pi_{j,} (\hat{\xi}_{t+1|T} (\div) \hat{\xi}_{t+1|t}).
\end{aligned}$$

Where $\Pi_{j,}$ is the jth row of Π. For the vector of probabilities one can therefore write:

$$\hat{\xi}_{t|T} = \hat{\xi}_{t|t} \odot \Pi (\hat{\xi}_{t+1|T} (\div) \hat{\xi}_{t+1|t}).$$

This is the earlier presented formula.

9.3 EM Algorithm for Autoregressive Processes with finite lag order

The here presented proofs for the equations (38), (39) and (40) closely follow Hamilton (1990, page 63-67). We begin with (38), then go on to (39) and finish with the proof for (40). The first, essential step for all three proofs, is to establish that the following holds:

$$\begin{aligned}
f_{\mathscr{Y}_{T:(m+1)}, \mathscr{S}|\mathscr{Y}_m; \lambda}(\vec{y}_{T:(m+1)}, \vec{s}_T|\vec{y}_m) = & f_{Y_T|Z_T;\alpha}(y_T|z_T) \cdot P_\Pi(S_T = s_T|S_{T-1} = s_{T-1}) \\
& \cdot f_{Y_{T-1}|Z_{T-1};\alpha}(y_{T-1}|z_{T-1}) \cdot P_\Pi(S_{T-1} = s_{T-1}|S_{T-2} = s_{T-2}) \\
& \cdot \ldots \\
& \cdot f_{Y_{m+1}|Z_{m+1};\alpha}(y_{m+1}|z_{m+1}) \cdot P_\Pi(S_{m+1} = s_{m+1}|S_m = s_m) \\
& \cdot \rho_{s_m, \ldots, s_1} \cdot
\end{aligned} \tag{55}$$

This can be derived in the following way:

$$
\begin{aligned}
& f_{Y_T|Z_T;\alpha}(y_T|z_T) \cdot P_\Pi(S_T = s_T|S_{T-1} = s_{T-1}) \\
& \cdot f_{Y_{T-1}|Z_{T-1};\alpha}(y_{T-1}|z_{T-1}) \cdot P_\Pi(S_{T-1} = s_{T-1}|S_{T-2} = s_{T-2}) \\
& \cdot ... \\
& \cdot f_{Y_{m+1}|Z_{m+1};\alpha}(y_{m+1}|z_{m+1}) \cdot P_\Pi(S_{m+1} = s_{m+1}|S_m = s_m) \\
& \cdot \rho_{s_m,...,s_1} \\
& = f_{Y_T|S_T,...,S_{T-m},Y_{T-1},...,Y_{T-m};\alpha}(y_T|s_T,...,s_{T-m},y_{T-1},...,y_{T-m}) P_\Pi(S_T = s_T|S_{T-1} = s_{T-1}) \\
& \quad \cdot f_{Y_{T-1}|S_{T-1},...,S_{T-1-m},Y_{T-1-1},...,Y_{T-1-m};\alpha}(y_{T-1}|s_{T-1},...,s_{T-1-m},y_{T-1-1},...,y_{T-1-m}) \\
& \quad \cdot P_\Pi(S_{T-1} = s_{T-1}|S_{T-2} = s_{T-2}) \\
& \quad \cdot ... \\
& \quad \cdot f_{Y_{m+2}|S_{m+2},...,S_2,Y_{m+1},...,Y_2;\alpha}(y_{m+2}|s_{m+2},...,s_2,y_{m+1},...,y_2) P_\Pi(S_{m+2} = s_{m+2}|S_{m+1} = s_{m+1}) \\
& \quad \cdot f_{Y_{m+1}|S_{m+1},...,S_1,Y_m,...,Y_1;\alpha}(y_{m+1}|s_{m+1},...,s_1,y_m,...,y_1) P_\Pi(S_{m+1} = s_{m+1}|S_m = s_m) \\
& \quad \cdot P_\lambda(S_m = s_m,...,S_1 = s_1|Y_m = y_m,...,Y_1 = y_1).
\end{aligned}
$$

And due to the Markov property and (12) it holds that:

$$
\begin{aligned}
& f_{Y_{m+1}|S_{m+1},...,S_1,Y_m,...,Y_1;\alpha}(y_{m+1}|s_{m+1},...,s_1,y_m,...,y_1) P_\Pi(S_{m+1} = s_{m+1}|S_m = s_m) \\
& = f_{Y_{m+1}|S_{m+1},...,S_1,Y_m,...,Y_1;\alpha}(y_{m+1}|s_{m+1},...,s_1,y_m,...,y_1) P_\Pi(S_{m+1} = s_{m+1}|S_m = s_m,...,S_1 = s_1) \\
& = f_{Y_{m+1}|S_{m+1},...,S_1,Y_m,...,Y_1;\alpha}(y_{m+1}|s_{m+1},...,s_1,y_m,...,y_1) \\
& \quad \cdot P_\Pi(S_{m+1} = s_{m+1}|S_m = s_m,...,S_1 = s_1,Y_m = y_m,...,Y_1 = y_1) \\
& = f_{Y_{m+1},S_{m+1}|S_m,...,S_1,Y_m,...,Y_1;\theta}(y_{m+1},s_{m+1}|s_m,...,s_1,y_m,...,y_1).
\end{aligned}
$$

Logically it also holds that:

$$
\begin{aligned}
& f_{Y_{m+1},S_{m+1}|S_m,...,S_1,Y_m,...,Y_1;\theta}(y_{m+1},s_{m+1}|s_m,...,s_1,y_m,...,y_1) \cdot P_\lambda(S_m = s_m,...,S_1 = s_1|Y_m = y_m,...,Y_1 = y_1) \\
& = f_{Y_{m+1},S_{m+1},S_m,...,S_1|Y_m,...,Y_1;\theta}(y_{m+1},s_{m+1},...,s_1|y_m,...,y_1).
\end{aligned}
$$

We assumed in our model formulation in 4.1 that there is a maximal autoregressive lag order m such that Y_t, depends only on m lags of Y_t. Then it holds that:

$$
\begin{aligned}
& f_{Y_{m+2}|S_{m+2},...,S_2,Y_{m+1},...,Y_2;\alpha}(y_{m+2}|s_{m+2},...,s_2,y_{m+1},...,y_2) P_\Pi(S_{m+2} = s_{m+2}|S_{m+1} = s_{m+1}) \\
& = f_{Y_{m+2}|S_{m+2},...,S_2,S_1,Y_{m+1},...,Y_2,Y_1;\alpha}(y_{m+2}|s_{m+2},...,s_2,s_1,y_{m+1},...,y_2,y_1) P_\Pi(S_{m+2} = s_{m+2}|S_{m+1} = s_{m+1}) \\
& = f_{Y_{m+2}|S_{m+2},...,S_2,S_1,Y_{m+1},...,Y_2,Y_1;\alpha}(y_{m+2}|s_{m+2},...,s_2,s_1,y_{m+1},...,y_2,y_1) \\
& \quad \cdot P_\Pi(S_{m+2} = s_{m+2}|S_{m+1} = s_{m+1},S_m = s_m,...,S_1 = s_1,Y_{m+1} = y_{m+1},...,Y_1 = y_1) \\
& = f_{Y_{m+2},S_{m+2}|S_{m+1},S_m,...,S_1,Y_{m+1},Y_m,...,Y_1;\theta}(y_{m+2},s_{m+2}|s_{m+1},s_m,...,s_1,y_{m+1},y_m,...,y_1).
\end{aligned}
$$

And thus we can write:

$$
\begin{aligned}
&f_{Y_{m+2},S_{m+2}|S_{m+1},S_m,...,S_1,Y_{m+1},Y_m,...,Y_1;\theta}(y_{m+2},s_{m+2}|s_{m+1},s_m,...,s_1,y_{m+1},y_m,...,y_1)\\
&\cdot f_{Y_{m+1},S_{m+1},S_m,...,S_1|Y_m,...Y_1;\theta}(y_{m+1},s_{m+1},s_m,...,s_1|y_m,...,y_1)\\
&= f_{Y_{m+2},S_{m+2}|S_{m+1},Y_{m+1},S_m,...,S_1,Y_m,...,Y_1;\theta}(y_{m+2},s_{m+2}|s_{m+1},y_{m+1},s_m,...,s_1,y_m,...,y_1)\\
&\quad \cdot f_{Y_{m+1},S_{m+1},S_m,...,S_1|Y_m,...Y_1;\theta}(y_{m+1},s_{m+1},s_m,...,s_1|y_m,...,y_1)\\
&= f_{Y_{m+2},S_{m+2},Y_{m+1},S_{m+1},S_m,...S_1|Y_m,...Y_1;\theta}(y_{m+2},s_{m+2},y_{m+1},s_{m+1},s_m,...,s_1|y_m,...,y_1).
\end{aligned}
$$

We can follow this logic until T and end up with:

$$
\begin{aligned}
&f_{Y_T|S_T,...,S_{T-m},Y_{T-1},...,Y_{T-m};\alpha}(y_T|s_T,...,s_{T-m},y_{T-1},...,y_{T-m})P_\Pi(S_T=s_T|S_{T-1}=s_{T-1})\\
&\cdot f_{Y_{T-1}|S_{T-1},...,S_{T-1-m},Y_{T-1-1},...,Y_{T-1-m};\alpha}(y_{T-1}|s_{T-1},...,s_{T-1-m},y_{T-1-1},...,y_{T-1-m})\\
&\cdot P_\Pi(S_{T-1}=s_{T-1}|S_{T-2}=s_{T-2})\\
&\cdot \dots\\
&\cdot f_{Y_{m+2}|S_{m+2},...,S_2,Y_{m+1},...,Y_2;\alpha}(y_{m+2}|s_{m+2},...,s_2,y_{m+1},...,y_2)P_\Pi(S_{m+2}=s_{m+2}|S_{m+1}=s_{m+1})\\
&\cdot f_{Y_{m+1}|S_{m+1},...,S_2,Y_m,...,Y_1;\alpha}(y_{m+1}|s_{m+1},...,s_2,y_m,...,y_1)P_\Pi(S_{m+1}=s_{m+1}|S_m=s_m)\\
&\cdot P_\lambda(S_m=s_m,...,S_1=s_1|Y_m=y_m,...,Y_1=y_1)\\
&= f_{Y_T,S_T,Y_{T-1},S_{T-1},...,Y_{m+1},S_{m+1},S_m,...,S_1|Y_m,...,Y_1;\lambda}(y_T,s_T,y_{T-1},s_{T-1},...,y_{m+1},s_{m+1},s_m,...,s_1|y_m,...,y_1)\\
&= f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\lambda}(\vec{y}_{T:(m+1)},\vec{s}_T|\vec{y}_m).
\end{aligned}
$$

Derivation of (38): Now that this first step has been established, we can now focus on the derivation of the first equation, thereby we are closely following Hamilton (1990, page 63-65). We start with (55), it holds that:

$$
\begin{aligned}
\ln(f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\lambda}(\vec{y}_{T:(m+1)},\vec{s}_T|\vec{y}_m)) = &\ln(f_{Y_T|Z_T;\alpha}(y_T|z_T))\\
&+\ln(P_\Pi(S_T=s_T|S_{T-1}=s_{T-1}))\\
&+\ln(f_{Y_{T-1}|Z_{T-1};\alpha}(y_{T-1}|z_{T-1}))\\
&+\ln(P_\Pi(S_{T_1}=s_{T-1}|S_{T-2}=s_{T-2}))\\
&+\dots\\
&+\ln(f_{Y_{m+1}|Z_{m+1};\alpha}(y_{m+1}|z_{m+1}))\\
&+\ln(P_\Pi(S_{m+1}=s_{m+1}|S_m=s_m))\\
&+\ln(\rho_{s_m,...,s_1}).
\end{aligned}
\tag{56}
$$

We remember that if we have $L(x,y)$ and $\ln(L(x,y))=l(x,y)$, then it is true that:

$$
\frac{\partial l(x,y)}{\partial x}=\frac{1}{L(x,y)}\frac{\partial L(x,y)}{\partial x},
$$

and thus

$$
\frac{\partial L(x,y)}{\partial x}=\frac{\partial l(x,y)}{\partial x}L(x,y).
$$

We apply this now:

$$f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\lambda}(\vec{y}_{T:(m+1)},\vec{s}_T|\vec{y}_m)\frac{\partial \ln(f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\lambda}(\vec{y}_{T:(m+1)},\vec{s}_T|\vec{y}_m))}{\partial \pi_{i,j}}$$
$$=\frac{\partial f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\lambda}(\vec{y}_{T:(m+1)},\vec{s}_T|\vec{y}_m)}{\partial \pi_{i,j}},$$

where:

$$\frac{\partial \ln(f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\lambda}(\vec{y}_{T:(m+1)},\vec{s}_T|\vec{y}_m))}{\partial \pi_{i,j}}=\sum_{t=m+1}^{T}\frac{\partial \ln(P_\Pi(S_t=s_t|S_{t-1}=s_{t-1}))}{\partial \pi_{i,j}}.$$

One should note that:

$$\frac{\partial \ln(P_\Pi(S_t=s_t|S_{t-1}=s_{t-1}))}{\partial \pi_{i,j}}=\begin{cases}\frac{1}{\pi_{i,j}}, & \text{if } S_t=j \quad \text{and} \quad S_{t-1}=i\\ 0, & \text{otherwise}\end{cases}.$$

In the following, we will use the Kronecker delta as notation in the following way:

$$\delta_{[A]}=\begin{cases}1, & \text{if A is true}\\ 0, & \text{otherwise}\end{cases},$$

thus:

$$\frac{\partial f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\lambda}(\vec{y}_{T:(m+1)},\vec{s}_T|\vec{y}_m)}{\partial \pi_{i,j}}=f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\lambda}(\vec{y}_{T:(m+1)},\vec{s}_T|\vec{y}_m)\sum_{t=m+1}^{T}\frac{\partial \ln(P_\Pi(S_t=s_t|S_{t-1}=s_{t-1}))}{\partial \pi_{i,j}}$$
$$=f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\lambda}(\vec{y}_{T:(m+1)},\vec{s}_T|\vec{y}_m)\frac{1}{\pi_{i,j}}\sum_{t=m+1}^{T}\delta_{[S_t=j,S_{t-1}=i]}.$$

We remember that the following holds:

$$Q_{\lambda_l,\vec{y}_T}(\lambda_{l+1})=\sum_{\vec{s}_T}\ln(f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\lambda_{l+1}}(\vec{y}_{T:(m+1)},\vec{s}_T|\vec{y}_m))f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\lambda_l}(\vec{y}_{T:(m+1)},\vec{s}_T|\vec{y}_m).$$

Therefore, we can say:

$$\frac{\partial Q_{\lambda_l,\vec{y}_T}(\lambda_{l+1})}{\partial \pi_{i,j}^{(l+1)}}=\sum_{\vec{s}_T}\frac{\partial \ln(f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\lambda_{l+1}}(\vec{y}_{T:(m+1)},\vec{s}_T|\vec{y}_m))}{\partial \pi_{i,j}^{(l+1)}}f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\lambda_l}(\vec{y}_{T:(m+1)},\vec{s}_T|\vec{y}_m)$$
$$=\sum_{\vec{s}_T}\frac{1}{\pi_{i,j}^{(l+1)}}\sum_{t=m+1}^{T}\delta_{[S_t=j,S_{t-1}=i]}f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\lambda_l}(\vec{y}_{T:(m+1)},\vec{s}_T|\vec{y}_m).$$

It is now essential to notice that:

$$\sum_{\vec{s}_T}\delta_{[S_t=j,S_{t-1}=i]}f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\lambda_l}(\vec{y}_{T:(m+1)},\vec{s}_T|\vec{y}_m)=f_{\mathscr{Y}_{T:(m+1)},S_t,S_{t-1}|\mathscr{Y}_m;\lambda_l}(\vec{y}_{T:(m+1)},j,i)$$
$$=P_{\lambda_l}(S_t=j,S_{t-1}=i|\mathscr{Y}_T=\vec{y}_T)$$
$$\cdot f_{\mathscr{Y}_{T:(m+1)}|\mathscr{Y}_m;\lambda_l}(\vec{y}_{T:(m+1)}|\vec{y}_m),$$

and that therefore:

$$\begin{aligned}\frac{\partial Q_{\lambda_l,\vec{y}_T}(\lambda_{l+1})}{\partial \pi_{i,j}^{(l+1)}} &= \sum_{\vec{s}_T} \frac{1}{\pi_{i,j}^{(l+1)}} \sum_{t=m+1}^{T} \delta_{[S_t=j,S_{t-1}=i]} f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\lambda_l}(\vec{y}_{T:(m+1)},\vec{s}_T|\vec{y}_m) \\ &= \frac{1}{\pi_{i,j}^{(l+1)}} \sum_{t=m+1}^{T} \sum_{\vec{s}_T} \delta_{[S_t=j,S_{t-1}=i]} f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\lambda_l}(\vec{y}_{T:(m+1)},\vec{s}_T|\vec{y}_m) \\ &= \frac{1}{\pi_{i,j}^{(l+1)}} \sum_{t=m+1}^{T} P_{\lambda_l}(S_t=j,S_{t-1}=i|\mathscr{Y}_T=\vec{y}_T) f_{\mathscr{Y}_{T:(m+1)}|\mathscr{Y}_m;\lambda_l}(\vec{y}_{T:(m+1)}|\vec{y}_m).\end{aligned}$$

Under the constraint $\sum_{j=1}^{N} \pi_{i,j} = 1$ we can now form the Lagrangian:

$$Q_{\lambda_l,\vec{y}_T}(\lambda_{l+1}) - \mu_i(\sum_{j=1}^{N} \pi_{i,j} - 1).$$

This leads to the following first-order conditons:

$$\frac{\partial Q_{\lambda_l,\vec{y}_T}(\lambda_{l+1})}{\partial \pi_{i,j}^{(l+1)}} = \mu_i, \quad \text{for} \quad j = 1,...,N.$$

We insert our result from above:

$$\begin{aligned}&\frac{1}{\pi_{i,j}^{(l+1)}} \sum_{t=m+1}^{T} P_{\lambda_l}(S_t=j,S_{t-1}=i|\mathscr{Y}_T=\vec{y}_T) f_{\mathscr{Y}_{T:(m+1)}|\mathscr{Y}_m;\lambda_l}(\vec{y}_{T:(m+1)}|\vec{y}_m) = \mu_i \\ &\Leftrightarrow \sum_{t=m+1}^{T} P_{\lambda_l}(S_t=j,S_{t-1}=i|\mathscr{Y}_T=\vec{y}_T) = \frac{\pi_{i,j}^{(l+1)}\mu_i}{f_{\mathscr{Y}_{T:(m+1)}|\mathscr{Y}_m;\lambda_l}(\vec{y}_{T:(m+1)}|\vec{y}_m)}.\end{aligned}$$

We now sum over $1,...,N$, which leads to:

$$\begin{aligned}&\sum_{j=1}^{N} \sum_{t=m+1}^{T} P_{\lambda_l}(S_t=j,S_{t-1}=i|\mathscr{Y}_T=\vec{y}_T) = \sum_{j=1}^{N} \frac{\pi_{i,j}^{(l+1)}\mu_i}{f_{\mathscr{Y}_{T:(m+1)}|\mathscr{Y}_m;\lambda_l}(\vec{y}_{T:(m+1)}|\vec{y}_m)} \\ &\Leftrightarrow \sum_{t=m+1}^{T} P_{\lambda_l}(S_{t-1}=i|\mathscr{Y}_T=\vec{y}_T) = \frac{\mu_i}{f_{\mathscr{Y}_{T:(m+1)}|\mathscr{Y}_m;\lambda_l}(\vec{y}_{T:(m+1)}|\vec{y}_m)}.\end{aligned}$$

If we now insert this in the result from above we get:

$$\begin{aligned}&\sum_{t=m+1}^{T} P_{\lambda_l}(S_t=j,S_{t-1}=i|\mathscr{Y}_T=\vec{y}_T) = \frac{\pi_{i,j}^{(l+1)}\mu_i}{f_{\mathscr{Y}_{T:(m+1)}|\mathscr{Y}_m;\lambda_l}(\vec{y}_{T:(m+1)}|\vec{y}_m)} = \pi_{i,j}^{(l+1)} \sum_{t=m+1}^{T} P_{\lambda_l}(S_{t-1}=i|\mathscr{Y}_T=\vec{y}_T) \\ &\Leftrightarrow \pi_{i,j}^{(l+1)} = \frac{\sum_{t=m+1}^{T} P_{\lambda_l}(S_t=j,S_{t-1}=i|\mathscr{Y}_T=\vec{y}_T)}{\sum_{t=m+1}^{T} P_{\lambda_l}(S_{t-1}=i|\mathscr{Y}_T=\vec{y}_T)},\end{aligned}$$

this concludes the derivation of (38).

Derivation of (39): With that, we get to the second equation. In the following we closely follow Hamilton (1990, page 65-66). Again, we start with (55), but this time we take the derivative in α:

$$\frac{\partial f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\lambda}(\vec{y}_{T:(m+1)},\vec{s}_T|\vec{y}_m)}{\partial \alpha} = f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\lambda}(\vec{y}_{T:(m+1)},\vec{s}_T|\vec{y}_m) \sum_{t=m+1}^{T} \frac{\partial \ln(f_{Y_t|Z_t;\alpha}(y_t|z_t))}{\partial \alpha}.$$

It is important to note here that $f_{Y_t|Z_t;\alpha}(y_t|z_t)$ depends on S_t through Z_t, because $Z_t = (S_t, S_{t-1}, ..., S_{t-m}, Y_{t-1}, Y_{t-2}, ..., Y_{t-m})$, but at most for the dates $t, ..., t-m$, thus:

$$\begin{aligned}
\frac{\partial Q_{\lambda_l,\vec{y}_T}(\lambda_{l+1})}{\partial \alpha_{l+1}} &= \frac{\partial}{\partial \alpha_{l+1}} \sum_{\vec{s}_T} \ln(f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\lambda_{l+1}}(\vec{y}_{T:(m+1)}, \vec{s}_T|\vec{y}_m)) f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\lambda_l}(\vec{y}_{T:(m+1)}, \vec{s}_T|\vec{y}_m) \\
&= \sum_{\vec{s}_T} \frac{\partial \ln(f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\lambda_{l+1}}(\vec{y}_{T:(m+1)}, \vec{s}_T|\vec{y}_m))}{\partial \alpha_{l+1}} f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\lambda_l}(\vec{y}_{T:(m+1)}, \vec{s}_T|\vec{y}_m) \\
&\overset{(56)}{=} \sum_{\vec{s}_T} f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\lambda_l}(\vec{y}_{T:(m+1)}, \vec{s}_T|\vec{y}_m) \sum_{t=m+1}^{T} \frac{\partial \ln(f_{Y_t|Z_t;\alpha_{l+1}}(y_t|z_t))}{\partial \alpha_{l+1}} \\
&= \sum_{t=m+1}^{T} \sum_{\vec{s}_T} \frac{\partial \ln(f_{Y_t|Z_t;\alpha_{l+1}}(y_t|z_t))}{\partial \alpha_{l+1}} f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\lambda_l}(\vec{y}_{T:(m+1)}, \vec{s}_T|\vec{y}_m) \\
&= \sum_{t=m+1}^{T} \sum_{s_t=1}^{N} \cdots \sum_{s_{t-m}=1}^{N} \frac{\partial \ln(f_{Y_t|Z_t;\alpha_{l+1}}(y_t|z_t))}{\partial \alpha_{l+1}} \\
&\quad \cdot \left(\sum_{s_T=1}^{N} \cdots \sum_{s_{t+1}=1}^{N} \sum_{s_{t-m-1}=1}^{N} \cdots \sum_{s_1=1}^{N} f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\lambda_l}(\vec{y}_{T:(m+1)}, \vec{s}_T|\vec{y}_m) \right) \\
&= \sum_{t=m+1}^{T} \sum_{s_t=1}^{N} \cdots \sum_{s_{t-m}=1}^{N} \frac{\partial \ln(f_{Y_t|Z_t;\alpha_{l+1}}(y_t|z_t))}{\partial \alpha_{l+1}} \\
&\quad \cdot P_{\lambda_l}(S_t = s_t, ..., S_{t-m} = s_{t-m}|Y_T = y_T, ..., Y_1 = y_1) \\
&\quad \cdot f_{Y_T,...,Y_{m+1}|Y_m,...,Y_1;\lambda_l}(y_T, ..., y_{m+1}|y_m, ..., y_1).
\end{aligned}$$

These steps are possible because $f_{Y_t|Z_t;\alpha}(y_t|z_t)$ at most only depends on $S_t, ..., S_{t-m}$. This leads us to the following first order condition:

$$\begin{aligned}
&f_{\mathscr{Y}_{T:(m+1)}|\mathscr{Y}_m;\lambda_l}(\vec{y}_{T:(m+1)}|\vec{y}_m) \sum_{t=m+1}^{T} \sum_{s_t=1}^{N} \cdots \sum_{s_{t-m}=1}^{N} \frac{\partial \ln(f_{Y_t|Z_t;\alpha_{l+1}}(y_t|z_t))}{\partial \alpha_{l+1}} \\
&\cdot P_{\lambda_l}(S_t = s_t, S_{t-1} = s_{t-1}, ..., S_{t-m} = s_{t-m}|\mathscr{Y}_T = \vec{y}_T) = 0,
\end{aligned}$$

which is equivalent to (39).

Derivation of (40): Now we can turn to equation number three, here we closely follow the derivations presented by Hamilton (1990, page 66-67). We start with (56) and take the derivative in $\rho_{i_m,...,i_1}$:

$$\frac{\partial \ln(f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\lambda_l}(\vec{y}_{T:(m+1)}, \vec{s}_T|\vec{y}_m))}{\partial \rho_{i_m,...,i_1}} = \frac{1}{\rho_{i_m,...,i_1}} \cdot \delta_{[S_m=i_m,...,S_1=i_1]},$$

because of this, it holds that:

$$\begin{aligned}
\frac{\partial Q_{\lambda_l,\vec{y}_T}(\lambda_{l+1})}{\partial \rho_{i_m,...,i_1}^{(l+1)}} &= \sum_{\vec{s}_T} \frac{\partial \ln(f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\lambda_{l+1}}(\vec{y}_{T:(m+1)}, \vec{s}_T|\vec{y}_m))}{\partial \rho_{i_m,...,i_1}^{(l+1)}} f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\lambda_l}(\vec{y}_{T:(m+1)}, \vec{s}_T|\vec{y}_m) \\
&= \sum_{\vec{s}_T} \frac{1}{\rho_{i_m,...,i_1}^{(l+1)}} \delta_{[S_m=i_m,...,S_1=i_1]} f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\lambda_l}(\vec{y}_{T:(m+1)}, \vec{s}_T|\vec{y}_m).
\end{aligned}$$

We want to optimize $Q_{\lambda_l,\vec{y}_T}(\lambda_{l+1})$ under the constraint $\sum_{j=1}^{N^m}(\rho_{l+1})_j = 1$, i.e that the sum of all elements of ρ_{l+1} shall be 1. Thus we construct the Lagrangian:

$$Q_{\lambda_l,\vec{y}_T}(\lambda_{l+1}) - \mu(\sum_{j=1}^{N^m}(\rho_{l+1})_j - 1)).$$

Which leads to the first order condition:

$$\begin{aligned}
&\sum_{\vec{s}_T} \frac{1}{\rho^{(l+1)}_{i_m,\dots,i_1}} \delta_{[S_m=i_m,\dots,S_1=i_1]} f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\lambda_l}(\vec{y}_{T:(m+1)},\vec{s}_T|\vec{y}_m) = \mu \\
&\Leftrightarrow \sum_{\vec{s}_T} \delta_{[S_m=i_m,\dots,S_1=i_1]} f_{\mathscr{Y}_{T:(m+1)},\mathscr{S}|\mathscr{Y}_m;\lambda_l}(\vec{y}_{T:(m+1)},\vec{s}_T|\vec{y}_m) = \rho^{(l+1)}_{i_m,\dots,i_1}\mu \\
&\Leftrightarrow f_{S_m,\dots,S_1,Y_T,\dots,Y_{m+1}|Y_m,\dots,Y_1;\lambda_l}(i_m,\dots,i_1,y_T,\dots,y_{m+1}|y_m,\dots,y_1) = \rho^{(l+1)}_{i_m,\dots,i_1}\mu \\
&\Leftrightarrow P_{\lambda_l}(S_m = i_m,\dots,S_1 = i_1|\mathscr{Y}_T = \vec{y}_T) f_{\mathscr{Y}_{T:(m+1)}|\mathscr{Y}_m;\lambda_l}(\vec{y}_{T:(m+1)}|\vec{y}_m) = \mu\rho^{(l+1)}_{i_m,\dots,i_1}.
\end{aligned}$$

If we now sum over all potential values of $(i_1,\dots i_m)$ we end up with:

$$\begin{aligned}
&\sum_{i_m=1}^{N}\cdots\sum_{i_1=1}^{N} P_{\lambda_l}(S_m = i_m,\dots,S_1 = i_1|\mathscr{Y}_T = \vec{y}_T) f_{\mathscr{Y}_{T:(m+1)}|\mathscr{Y}_m;\lambda_l}(\vec{y}_{T:(m+1)}|\vec{y}_m) = \sum_{i_m=1}^{N}\cdots\sum_{i_1=1}^{N}\mu\rho^{(l+1)}_{i_m,\dots,i_1} \\
&\Leftrightarrow f_{\mathscr{Y}_{T:(m+1)}|\mathscr{Y}_m;\lambda_l}(\vec{y}_{T:(m+1)}|\vec{y}_m) = \mu.
\end{aligned}$$

We insert this for μ and get:

$$\begin{aligned}
&P_{\lambda_l}(S_m = i_m,\dots,S_1 = i_1|\mathscr{Y}_T = \vec{y}_T) f_{\mathscr{Y}_{T:(m+1)}|\mathscr{Y}_m;\lambda_l}(\vec{y}_{T:(m+1)}|\vec{y}_m) = f_{\mathscr{Y}_{T:(m+1)}|\mathscr{Y}_m;\lambda_l}(\vec{y}_{T:(m+1)}|\vec{y}_m)\rho^{(l+1)}_{i_m,\dots,i_1} \\
&\Leftrightarrow P_{\lambda_l}(S_m = i_m,\dots,S_1 = i_1|\mathscr{Y}_T = \vec{y}_T) = \rho^{(l+1)}_{i_m,\dots,i_1}.
\end{aligned}$$

This concludes the derivation of the EM algorithm for models with an underlying Markov-Chain and a dependence on a maximum lag order.

9.4 Stress Testing MSARM and MSwM: Results of 288 Random Processes

In section 7, we presented the results of six simulations comparing MSARM and MSwM. While these already demonstrated that MSwM can fail to accurately estimate certain processes where MSARM succeeds, we wanted to ensure that these findings were not driven by specific process choices. Therefore, we conducted an additional simulation study, randomly generating 48 time series for the Examples 0 to 5. To ensure robustness, the time series were generated with varying sample sizes: 50, 100, 150, 200, 250, 300, 350 and 400 observations (six series per sample size). Each process featured exactly two regimes, while the AR lag order was randomly drawn from a uniform distribution between 1 and 4. For Example 4 and 5, the switching vector was randomly generated. In Example 4, an additional check ensured that at least two parameters switched. The transition matrices were constructed such that the probability of remaining in the same state ranged between 90% and 99.5%. Intercepts were drawn from a normal distribution with $\mu = 0$ and $\sigma^2 = 25$. AR coefficients were constructed such that ϕ_k were sampled from uniform distributions over $\left(\frac{-1}{k+1}, \frac{1}{k+1}\right)$, and the error term standard deviation was drawn from a uniform distribution over $(0.5, 3)$. For each case where either MSARM

or MSwM failed (defined as a misclassification rate above 25% or an APaEE above 0.5), we additionally applied a Ljung-Box test at the 5% level to assess whether the generated time series exhibited statistically significant autocorrelation. The following graphic and tables summarize the results for Examples 0 to 5. In the graphic, the blue areas highlight cases where MSARM outperformed MSwM (lower MCR). The tables indicate whether MSARM or MSwM failed and whether the Ljung-Box test rejected the white noise hypothesis. Throughout the discussion, we state that it "would have been better to choose package Y instead of package X" whenever one package delivered a valid estimation while the other did not.

Example 0: For Example 0 we find that both MSARM and MSwM perform relatively similar, whereby MSARM performs slightly better, as there are 14 cases where MSARM fails to fulfill the quality criteria set by us, while MSwM fails 20 times to fulfill our criteria. Out of the 14 times MSARM failed, 11 times MSwM failed too. Therefore out of 48 randomly generated processes only 3 times it would have been better to use MSwM instead of MSARM (with standard settings), but it would have been in 9 cases better to use MSARM instead of MSwM. Therefore, one obtains a ratio of 3:1 in favor of MSARM. This is also reflected in Figure 16, where it becomes clear, that MSARM slightly tends to outperform MSwM.

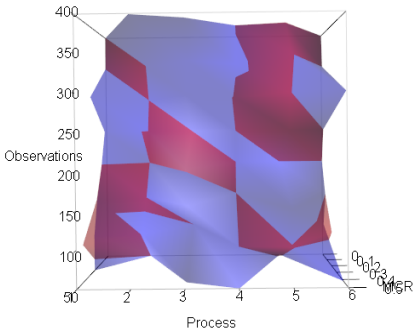


Figure 16: Example 0: 48 Random Processes

	50	**100**	**150**	**200**	**250**	**300**	**350**	**400**
Process 1	MSwM (TRUE)	MSARM (FALSE)	NA NA	NA NA	MSARM (TRUE)	MSwM (TRUE)	NA NA	NA NA
Process 2	NA NA	NA NA	NA NA	NA NA	NA NA	NA NA	NA NA	Both (TRUE)
Process 3	Both (TRUE)	NA NA	MSwM (TRUE)	Both (TRUE)	NA NA	MSwM (TRUE)	MSwM (TRUE)	MSwM (TRUE)
Process 4	Both (TRUE)	Both (FALSE)	NA NA	NA NA	NA NA	NA NA	NA NA	Both (TRUE)
Process 5	MSARM (TRUE)	Both (FALSE)	MSwM (TRUE)	NA NA	Both (TRUE)	Both (TRUE)	Both (FALSE)	Both (TRUE)
Process 6	MSwM (TRUE)	NA NA	NA NA	NA NA	NA NA	MSwM (TRUE)	NA NA	NA NA

Table 19: Example 0: Package Failure and Ljung-Box Test Results

Example 1: For Example 1 we find that MSARM tends to outperform MSwM. There are 18 cases where MSARM fails and 25 cases where MSwM fails. Out of the 18 cases where MSARM failed, MSwM failed in 17 cases too. Thus, there was only 1 case were one would have been better off choosing MSwM instead of MSARM (with standard settings). Meanwhile, there were 8 cases where one would have been better off choosing MSARM instead of MSwM. Therefore, one obtains a ratio of 8:1 in favor of MSARM, which is clearly reflected in Figure 17. Additionally, it should be noted that in the single case where it would have been better to choose MSwM over MSARM, it was not possible to reject the hypothesis that the underlying process is white noise, indicating a rather difficult estimation setup.

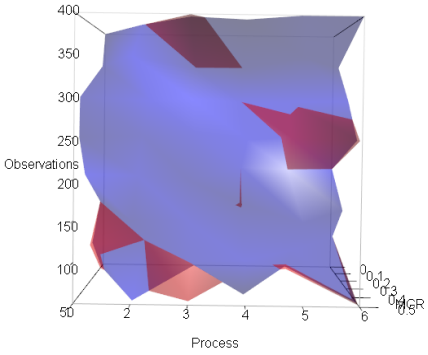


Figure 17: Example 1: 48 Random Processes

	50	**100**	**150**	**200**	**250**	**300**	**350**	**400**
Process 1	NA NA	NA NA	NA NA	Both (TRUE)	Both (TRUE)	MSwM (TRUE)	NA NA	NA NA
Process 2	Both (TRUE)	NA NA	MSwM (FALSE)	Both (TRUE)	NA NA	NA NA	Both (FALSE)	Both (TRUE)
Process 3	MSARM (FALSE)	Both (TRUE)	Both (TRUE)	MSwM (TRUE)	NA NA	MSwM (TRUE)	NA NA	Both (TRUE)
Process 4	Both (TRUE)	Both (TRUE)	NA NA	NA NA	Both (TRUE)	NA NA	NA NA	Both (TRUE)
Process 5	NA NA	MSwM (TRUE)	MSwM (TRUE)	NA NA	NA NA	NA NA	Both (TRUE)	Both (TRUE)
Process 6	Both (FALSE)	NA NA	NA NA	NA NA	Both (TRUE)	MSwM (FALSE)	NA NA	MSwM (TRUE)

Table 20: Example 1: Package Failure and Ljung-Box Test Results

Example 2: For Example 2 we find an even extremer case of MSARM outperforming MSwM. Out of the 48 processes, there were only 2 cases where MSwM managed to estimate the underlying process reasonably well. There was only 1 case where it would have been better to choose MSwM over MSARM, while in 25 cases it would have been better to choose MSARM over MSwM, a ratio of 25:1 in favor of MSARM. Furthermore, it should be noted that again, the only case where MSwM would have been superior to MSARM was for a time series where the hypothesis that the underlying process is just white noise was not rejected.

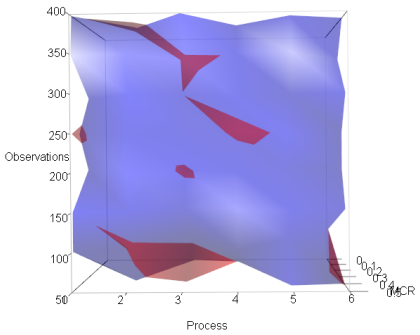


Figure 18: Example 2: 48 Random Processes

	50	**100**	**150**	**200**	**250**	**300**	**350**	**400**
Process 1	MSwM (TRUE)	MSwM (TRUE)	Both (TRUE)	MSwM (TRUE)	Both (TRUE)	MSwM (TRUE)	MSwM (TRUE)	Both (TRUE)
Process 2	Both (TRUE)	Both (TRUE)	MSwM (TRUE)	MSwM (TRUE)	MSwM (TRUE)	MSwM (TRUE)	MSwM (TRUE)	Both (TRUE)
Process 3	Both (FALSE)	MSwM (TRUE)	MSwM (TRUE)	Both (TRUE)	Both (TRUE)	Both (TRUE)	Both (TRUE)	MSwM (TRUE)
Process 4	NA NA	Both (TRUE)	Both (TRUE)	Both (TRUE)	Both (FALSE)	MSwM (TRUE)	Both (TRUE)	MSwM (TRUE)
Process 5	MSwM (TRUE)	Both (TRUE)	Both (TRUE)	MSwM (TRUE)	MSwM (TRUE)	MSwM (TRUE)	MSwM (TRUE)	Both (TRUE)
Process 6	Both (TRUE)	MSARM (FALSE)	MSwM (TRUE)	Both (TRUE)	MSwM (TRUE)	MSwM (TRUE)	MSwM (TRUE)	MSwM (TRUE)

Table 21: Example 2: Package Failure and Ljung-Box Test Results

Example 3: Example 3 again shows that MSARM tends to perform better than MSwM, especially regarding more general setups. This is the first example were the error term variance is allowed to switch and it turns out that there was not a single case, where it would have been better to use MSwM instead of MSARM. Additionally one should note that there were 12 cases where MSwM failed to meet our criteria, while MSARM was capable of doing so.

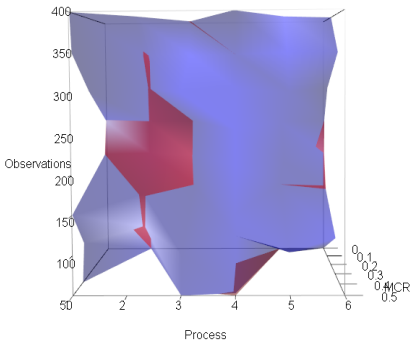


Figure 19: Example 3: 48 Random Processes

	50	100	150	200	250	300	350	400
Process 1	Both (TRUE)	Both (TRUE)	MSwM (TRUE)	NA NA	NA NA	MSwM (TRUE)	MSwM (TRUE)	MSwM (TRUE)
Process 2	NA NA	NA NA	NA NA	NA NA	NA NA	NA NA	NA NA	Both (TRUE)
Process 3	Both (TRUE)	MSwM (TRUE)	NA NA	NA NA	NA NA	Both (TRUE)	NA NA	NA NA
Process 4	Both (TRUE)	Both (FALSE)	MSwM (TRUE)	MSwM (TRUE)	MSwM (TRUE)	NA NA	NA NA	Both (TRUE)
Process 5	NA NA	Both (TRUE)	NA NA	MSwM (TRUE)	NA NA	Both (TRUE)	Both (FALSE)	MSwM (TRUE)
Process 6	NA NA	Both (FALSE)	NA NA	NA NA	NA NA	NA NA	MSwM (TRUE)	MSwM (TRUE)

Table 22: Example 3: Package Failure and Ljung-Box Test Results

Example 4: For Example 4 we find similar results, again MSARM tends to perform better than MSwM, there is only 1 case where MSARM did not meet the quality criteria while MSwM did. Meanwhile, there are 10 cases where it would have been better to utilize MSARM instead of MSwM, leading to a ratio of 10:1 in favor of MSARM.

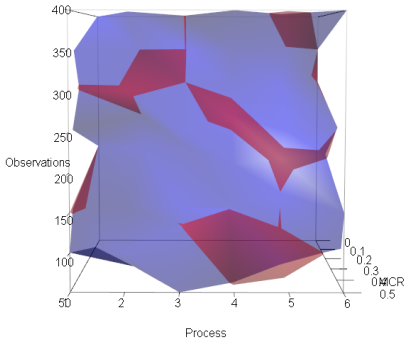


Figure 20: Example 4: 48 Random Processes

	50	100	150	200	250	300	350	400
Process 1	NA NA	MSwM (TRUE)	Both (TRUE)	NA NA	MSwM (TRUE)	Both (TRUE)	MSwM (TRUE)	NA NA
Process 2	Both (TRUE)	Both (TRUE)	Both (TRUE)	NA NA	NA NA	Both (TRUE)	NA NA	Both (TRUE)
Process 3	Both (FALSE)	Both (TRUE)	MSwM (TRUE)	NA NA	Both (TRUE)	NA NA	NA NA	NA NA
Process 4	Both (FALSE)	NA NA	Both (TRUE)	MSwM (TRUE)	Both (TRUE)	Both (TRUE)	MSwM (TRUE)	MSwM (TRUE)
Process 5	Both (TRUE)	NA NA	MSwM (TRUE)	Both (TRUE)	Both (TRUE)	MSwM (TRUE)	Both (TRUE)	Both (TRUE)
Process 6	Both (TRUE)	Both (TRUE)	MSwM (TRUE)	Both (TRUE)	Both (FALSE)	MSARM (TRUE)	NA NA	Both (TRUE)

Table 23: Example 4: Estimation Results and Ljung-Box Test Outcomes

Example 5: Last, but not least we find for Example 5 similar results to the previous Examples. We have 1 case, were MSARM did not reach our quality criteria, while MSwM did, but there were 6 cases

were MSwM failed to meet the criteria, while MSARM did. Therefore, we get a ratio of 6:1, in favor of MSARM.

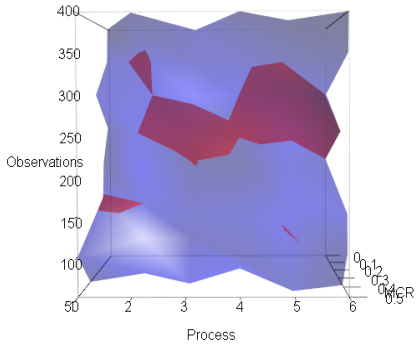


Figure 21: Example 5: 48 Random Processes

	50	**100**	**150**	**200**	**250**	**300**	**350**	**400**
Process 1	Both (TRUE)	Both (TRUE)	NA NA	NA NA	NA NA	NA NA	NA NA	Both (TRUE)
Process 2	Both (TRUE)	MSARM (TRUE)	Both (TRUE)	Both (TRUE)	Both (TRUE)	NA NA	NA NA	Both (TRUE)
Process 3	Both (FALSE)	MSwM (TRUE)	Both (TRUE)	NA NA	Both (TRUE)	Both (TRUE)	MSwM (TRUE)	NA NA
Process 4	Both (TRUE)	MSwM (TRUE)	Both (TRUE)	Both (TRUE)	NA NA	NA NA	NA NA	Both (TRUE)
Process 5	Both (TRUE)	NA NA	MSwM (TRUE)	Both (TRUE)	Both (TRUE)	Both (TRUE)	NA NA	Both (TRUE)
Process 6	Both (TRUE)	MSwM (TRUE)	Both (TRUE)	NA NA	Both (TRUE)	NA NA	MSwM (TRUE)	Both (TRUE)

Table 24: Example 5: Estimation Results and Ljung-Box Test Outcomes

Affidavit

Ich versichere, dass ich die vorliegende Arbeit ohne Hilfe Dritter und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt und die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen. Ich bin damit einverstanden, dass meine Arbeit zum Zwecke eines Plagiatsabgleichs in elektronischer Form anonymisiert versendet und gespeichert werden kann.

I affirm that this Bachelor thesis was written by myself without any unauthorised third-party support. All used references and resources are clearly indicated. All quotes and citations are properly referenced. This thesis was never presented in the past in the same or similar form to any examination board. I agree that my thesis may be subject to electronic plagiarism check. For this purpose, an anonymous copy may be distributed and uploaded to servers within and outside the University of Mannheim.

Place, Date: Mannheim, 04.07.2025 **Signature:** 