Hypothesis Testing in βARMA Models: An Example with R

Illustrating Likelihood Ratio, Score, and Wald Tests based on Costa, Cribari-Neto and Scher (2024)

Everton da Costa

2025-06-27

Contents

1 Load functions **1**

2 Introduction **1**

3 Data Simulation **2**

4 Parameter Estimation **3**

5 Classical Hypothesis Tests: LR, Rao Score, and Wald **4**

5.1 Likelihood Ratio Test 4

5.2 Rao Score Test 5

5.3 Wald Test 6

1 Load functions

```
source(here::here("R", "simuBarma.R"))
source(here::here("R", "makeLinkStructure.R"))
source(here::here("R", "barma3ClassicalTestsAuxFun.R"))
source(here::here("R", "barma3ClassicalTests.R"))

# For reproducibility of the example from Costa, Cribari-Neto and Scher (2024).
seed <- 2
link_function <- "logit"
```

2 Introduction

This report demonstrates the application of classical hypothesis tests—Wald, Likelihood Ratio (LR), and Rao Score tests—for Beta Autoregressive Moving Average (βARMA) models using R. The methodologies and interpretations are grounded in the research by Costa, Cribari-Neto, and Scher (2024). Their work critically examines the performance of these tests and highlights potential inaccuracies, especially when the null and alternative (unrestricted) models involve conditioning on a different number of initial observations.

Key Reference:

Costa, E., Cribari-Neto, F., & Scher, V. T. (2024). Test inferences and link function selection in dynamic beta modeling of seasonal hydro-environmental time series with temporary abnormal regimes. Journal of Hydrology, 638, 131489. https://doi.org/10.1016/j.jhydrol.2024.131489

In βARMA models, the conditional log-likelihood sum starts from $t = a + 1$, where $a = \max(p, q)$ with p and q being the AR and MA orders, respectively. A crucial issue arises if a for the null model (a_N) is less than a for the non-null/unrestricted model (a_{NN}). This report will illustrate such a scenario ($a_N < a_{NN}$) and show how to apply the tests correctly. The numerical results presented here replicate the example in Section 3.1 of Costa, Cribari-Neto and Scher (2024).

3 Data Simulation

We simulate a time series of length $n = 250$ from a βAR(1) process. The true parameters are:

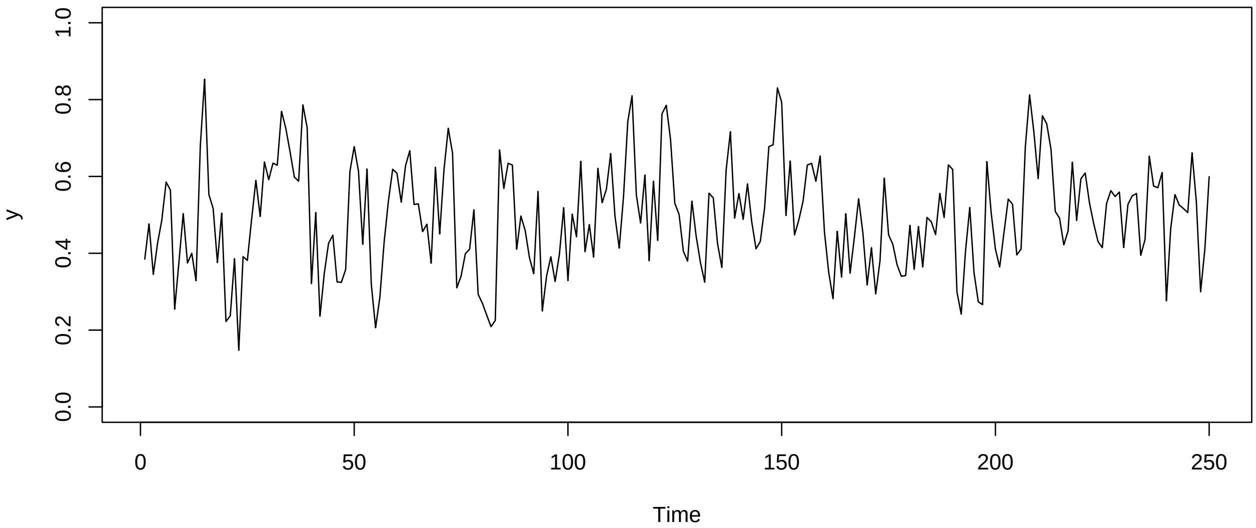
$\alpha = 0$, $\varphi_1 = 0.4$, $\phi = 20$, A logit link function is assumed for $g(\mu_t)$.

```
n <- 250
alpha_true <- 0
varphi_true <- 0.4
theta_true <- NA # Implies no MA part in simuBarma
phi_true <- 20

set.seed(seed) # Ensure reproducibility for this specific simulation
y <- simuBarma(
  n = n,
  alpha = alpha_true,
  varphi = varphi_true,
  theta = theta_true,
  phi = phi_true,
  link = link_function
)

plot(as.numeric(y),
  xaxt = "n", yaxt = "n",
  xlab = "Time", ylab = "y",
  ylim = c(0, 1),
  type = "l",
)

# Adjust the axis values.
axis(1, seq(0, n, by = 50), )
axis(2, seq(0, 1, 0.2))
```



4 Parameter Estimation

We fit an unrestricted βARMA(1,4) model and a restricted βARMA(1,0) model. Here, $a_{NN} = \max(1,4) = 4$ for the unrestricted model, and $a_N = \max(1,0) = 1$ for the restricted model.

```
ar_vec <- 1
ma_vec <- 1:4
rest_ma <- 3:6

fit <- barma3ClassicalTests(
  y = y,
  ar = ar_vec,
  ma = ma_vec,
  rest_ma = rest_ma
)

fit_unrest <- fit$unrestricted_model
fit_rest <- fit$ar_restricted_model

unrest_coeff <- round(fit_unrest$coeff, 4)
rest_coeff <- round(fit_rest$coeff, 4)

# Unrestricted Model BARMA(1,4) Coefficients:
print(unrest_coeff)
```

```
##   alpha varphi1  theta1  theta2  theta3  theta4     phi
## -0.0102  0.3936  0.0564  0.0319  0.0019  0.0346 15.8319
```

```
# Restricted Model BARMA(1,0) Coefficients:
print(rest_coeff)
```

```
##   alpha varphi1     phi
## -0.0108  0.4523 15.9138
```

5 Classical Hypothesis Tests: LR, Rao Score, and Wald

This section details the application of the Wald, Likelihood Ratio (LR), and Rao Score tests. All three tests are employed to evaluate a common hypothesis regarding the parameters of the βARMA model. Specifically, we test the joint significance of the first four moving average (MA) parameters by comparing an unrestricted βARMA(1,4) model against a restricted βARMA(1,0) model.

The null hypothesis: $H_0 : \theta_1 = \theta_2 = \theta_3 = \theta_4 = 0$. The alternative hypothesis, H_1, is that at least one of these θ_j parameters (for $j = 1, 2, 3, 4$) is non-zero.

Under the null hypothesis, the test statistics for the Wald, LR, and Rao Score tests asymptotically follow a χ^2 (chi-squared) distribution. Since there are four parameters being restricted under H_0 (namely $\theta_1, \theta_2, \theta_3, \theta_4$), the degrees of freedom for this χ^2 distribution is 4.

A key focus of this report, following the findings of Costa, Cribari-Neto and Scher (2024), is addressing the methodological challenge that arises when the number of initial observations used for conditioning differs between the null model ($a_N = \max(1, 0) = 1$) and the unrestricted model ($a_{NN} = \max(1, 4) = 4$). We will demonstrate both standard and adjusted versions of these tests designed for such $a_N < a_{NN}$ scenarios to ensure robust inference.

5.1 Likelihood Ratio Test

The `barma3ClassicalTests` function provides three versions of the LR statistic to address this:

- LR1 (Naive): The conventional LR statistic, which may be unreliable in the current $a_N < a_{NN}$ scenario.
- LR2 (Adjusted): An LR statistic with an adjustment to the restricted model's log-likelihood to better align the comparison with the unrestricted model.
- LR3 (Recommended): The LR statistic specifically recommended by Costa, Cribari-Neto and Scher (2024) for situations like ours, $a_N < a_{NN}$.

The R code below presents these three statistics.

```r
# Classical tests results
classical_tests_results <- fit$classical_tests
```

```r
# Extract all LR statistics and p-values
LR1_stat <- classical_tests_results$LR_naive_res[1]
LR1_pval <- classical_tests_results$LR_naive_res[2]

LR2_stat <- classical_tests_results$LR_m0_res[1]
LR2_pval <- classical_tests_results$LR_m0_res[2]

LR3_stat <- classical_tests_results$LR_mfun_res[1]
LR3_pval <- classical_tests_results$LR_mfun_res[2]

# LR1 (Naive):
# Potentially unreliable when a_N < a_NN.
cat(paste0("LR1 (Naive) Statistic: ", round(LR1_stat, 2),
           " (p-value: ", round(LR1_pval, 4), ")\n"))
```

```
## LR1 (Naive) Statistic: -5.34 (p-value: 1)
```

```r
# LR2 (Adjusted):
# Adjusted for differing values between a_N and a_NN.
cat(paste0("LR2 (Adjusted) Statistic: ", round(LR2_stat, 2),
           " (p-value: ", round(LR2_pval, 4), ")\n"))
```

```
## LR2 (Adjusted) Statistic: 0.31 (p-value: 0.9892)
```

```r
# LR3 (Recommended):
# Preferred method for a_N < a_NN, ensuring comparable likelihood conditioning.
cat(paste0("LR3 (Recommended) Statistic: ", round(LR3_stat, 2),
           " (p-value: ", round(LR3_pval, 4), ")\n"))
```

```
## LR3 (Recommended) Statistic: 0.3 (p-value: 0.9898)
```

5.2 Rao Score Test

The Rao Score test, provides another way to test hypotheses, requiring only the estimation of the restricted model. It examines the gradient (score) of the log-likelihood function at the restricted estimates. Similar to the LR test, adjustments are crucial when $a_N < a_{NN}$ to ensure accurate inference, as discussed by Costa, Cribari-Neto and Scher (2024). The output below presents:

Score Vectors:

- `ar_score_vec_arma_naive`: The score vector, with parameters estimated using its own a_N.
- `ar_score_vec_arma_mfun`: The score vector, with parameters estimated using a_{NN}.

Score Test Statistics:

Naive Versions (using score_vec_naive and mat_vcov_naive from model with a_N):

- `R_naive_res` (Sr*): A form of the Score statistic that can exhibit size distortions in the $a_N < a_{NN}$ case.
- `R_expanded_res` (Se*): The the Score statistic, also using a_N.

Recommended Versions:

- `R_mfun_res` (Sr): The reduced form of the Score statistic, with parameters estimated using a_{NN}.
- `R_exp_mfun_res` (Se): The expanded form of the Score statistic, with parameters estimated using a_{NN}.

The R code below displays these components. For detailed theoretical justifications and the precise nature of the 'mfun' object, readers should consult Costa, Cribari-Neto and Scher (2024).

```r
# Extract all Score statistics and p-values
Sr_star_stat <- classical_tests_results$R_naive_res[1]
Sr_star_pval <- classical_tests_results$R_naive_res[2]

Se_star_stat <- classical_tests_results$R_expanded_res[1]
Se_star_pval <- classical_tests_results$R_expanded_res[2]

Sr_stat <- classical_tests_results$R_mfun_res[1]
Sr_pval <- classical_tests_results$R_mfun_res[2]

Se_stat <- classical_tests_results$R_exp_mfun_res[1]
Se_pval <- classical_tests_results$R_exp_mfun_res[2]

cat("Score Vector (a_N):\n")
```

```
## Score Vector (a_N):
```

```r
print(round(fit$ar_restricted_model$ar_score_vec_arma_naive, 4))
```

```
## [1]  1.6660  0.1062 -0.0165  0.6977 -3.5366  8.3800 -0.0528
```

```r
cat("Score Vector (a_NN):\n")
```

```
## Score Vector (a_NN):
```

```
print(round(fit$ar_restricted_model$ar_score_vec_arma_mfun, 4))
```

```
## [1]  0.0000  0.0000 -0.1286  0.6518 -3.5464  8.3098  0.0000
```

```
# Score Statistics (based on restricted model with a_N).
# Potentially less reliable if a_N < a_NN.
cat(paste0("Score Statistic (Sr*, Naive Reduced): ", round(Sr_star_stat, 4),
           " (p-value: ", round(Sr_star_pval, 4), ")\n"))
```

```
## Score Statistic (Sr*, Naive Reduced): 0.4343 (p-value: 0.9796)
```

```
cat(paste0("Score Statistic (Se*, Naive Extended): ", round(Se_star_stat, 4),
           " (p-value: ", round(Se_star_pval, 4), ")\n"))
```

```
## Score Statistic (Se*, Naive Extended): 0.3618 (p-value: 0.9855)
```

```
# Score Statistics (Recommended for a_N < a_NN).
cat(paste0("  Score Statistic (Sr, Recommended Reduced): ", round(Sr_stat, 4),
           " (p-value: ", round(Sr_pval, 4), ")\n"))
```

```
##   Score Statistic (Sr, Recommended Reduced): 0.3518 (p-value: 0.9862)
```

```
cat(paste0("  Score Statistic (Se, Recommended Extended): ", round(Se_stat, 4),
           " (p-value: ", round(Se_pval, 4), ")\n"))
```

```
##   Score Statistic (Se, Recommended Extended): 0.3518 (p-value: 0.9862)
```

5.3 Wald Test

The Wald test is another method for testing hypotheses about the model parameters. It primarily uses information from the unrestricted model, specifically the parameter estimates and their variance-covariance matrix, to assess the specified null hypothesis ($H_0 : \theta_1 = \theta_2 = \theta_3 = \theta_4 = 0$).

- W_1: This is the conventional Wald statistic, which employs the variance-covariance matrix obtained from the unrestricted βARMA(1,4) model.
- W_2: This version, also discussed by Costa, Cribari-Neto and Scher (2024). uses an alternative way to estimate the information matrix involved in the test construction.

```
# Extract Wald statistics and p-values
W1_stat <- classical_tests_results$W_res[1]
W1_pval <- classical_tests_results$W_res[2]

W2_stat <- classical_tests_results$W2_res[1]
W2_pval <- classical_tests_results$W2_res[2]

# W1 Statistic (Standard Wald Test).
cat(paste0("W1 (Standard) Statistic: ", round(W1_stat, 2),
           " (p-value: ", round(W1_pval, 4), ")\n"))
```

```
## W1 (Standard) Statistic: 0.29 (p-value: 0.9902)
```

```
# W2 Statistic.
cat(paste0("W2 (Adjusted) Statistic: ", round(W2_stat, 2),
           " (p-value: ", round(W2_pval, 4), ")\n"))
```

```
## W2 (Adjusted) Statistic: 0.27 (p-value: 0.9917)
```