Journal of Computational and Graphical Statistics

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/ucgs20

Frequency Domain Calculation of Seasonal VARMA Autocovariances

Tucker S. McElroy

To cite this article: Tucker S. McElroy (2022) Frequency Domain Calculation of Seasonal VARMA Autocovariances, Journal of Computational and Graphical Statistics, 31:1, 301-303, DOI: 10.1080/10618600.2021.1981343

To link to this article: https://doi.org/10.1080/10618600.2021.1981343

Published online: 12 Nov 2021.

JOURNAL OF COMPUTATIONAL AND GRAPHICAL STATISTICS
2022, VOL. 31, NO. 1, 301–303
https://doi.org/10.1080/10618600.2021.1981343

SHORT TECHNICAL NOTE

Frequency Domain Calculation of Seasonal VARMA Autocovariances

Tucker S. McElroy

U.S. Census Bureau, Research and Methodology Directorate, Washington, DC

ABSTRACT
In applications of the fitting and forecasting of Seasonal Vector AutoRegressive Moving Average (SVARMA) models, it is important to quickly and accurately compute the autocovariances. Recursive time domain approaches rely on expressing the seasonal AutoRegressive matrix polynomials as a high order non-seasonal AutoRegressive matrix polynomial; initialization of the recursions is therefore costly, because a large-dimensional matrix must be inverted. However, the resulting high-order polynomial has many zeroes that are not intelligently exploited, indicating that this time domain approach is not optimal. It is proposed to compute the autocovariances via integrating the spectral density, taking advantage of analytical expressions for the inverse AutoRegressive matrix polynomials in terms of determinant and adjoint. The R and RCPP code can be hundreds of times faster than the time domain methods when the dimension and seasonal period are large.

ARTICLE HISTORY
Received March 2021
Revised September 2021

KEYWORDS
Adjoint polynomial; Spectral density; Vector time series

1. Introduction

Consider an N-dimensional stationary time series $\{X_t\}$ that is a Seasonal Vector AutoRegressive Moving Average process. By definition this means there exist matrix polynomials $\phi(z)$, $\Phi(z)$, $\theta(z)$, and $\Theta(z)$ such that

$$\phi(B)\Phi(B^s)X_t = \theta(B)\Theta(B^s)Z_t,$$

where s is an integer seasonal period and $\{Z_t\}$ is a vector white-noise process of covariance matrix Σ. (See Lütkepohl 2005 for background, p. 419 and following.) The matrix polynomials have degrees p, P, q, and Q, respectively, and are given by

$$\phi(z) = I_N - \sum_{j=1}^{p} \phi_j z^j \qquad \Phi(z) = I_N - \sum_{j=1}^{P} \Phi_j z^j$$
$$\theta(z) = I_N - \sum_{j=1}^{q} \theta_j z^j \qquad \Theta(z) = I_N - \sum_{j=1}^{Q} \Theta_j z^j,$$

where each of the coefficients is a real $N \times N$ matrix, and I_N is the $N \times N$-dimensional identity matrix. There are stability conditions on these matrix polynomials, as described in Roy et al. (2019). For such a SVARMA process the autocovariances are well-defined, being given by $\mathrm{cov}[X_{t+h}, X_t] = \Gamma_h$ for $h \in \mathbb{Z}$. When fitting a SVARMA model via maximum likelihood, or when forecasting from an SVARMA model, it is important to be able to swiftly compute the autocovariances from the parameters.

Time domain approaches to this problem use recursive relations of the coefficients, as described in McElroy (2017). In particular, note that $\varphi(z) = \phi(z)\Phi(z^s)$ defines a matrix polynomial of degree $p + Ps$. Similarly, setting $\vartheta(z) = \theta(z)\Theta(z^s)$ (which has degree $q + Qs$), we obtain a nonseasonal VARMA of the form $\varphi(B)X_t = \vartheta(B)Z_t$, to which standard time domain methods can be applied. However, a matrix inversion is required to initialize the recursions, and the matrix dimension is given by the degree of the VAR times the cross-section, that is, a $(p + Ps)N \times (p + Ps)N$-dimensional matrix must be inverted. For monthly or quarterly data ($s = 4, 12$), this is not onerous, but for weekly data ($s = 52$) the inversion can be prohibitively burdensome.

This article provides a new approach based on using the frequency domain, noting that autocovariances can be computed as an integral of the spectral density. However, the SVARMA spectral density requires the inversion of $\phi(e^{-i\lambda})$ and $\Phi(e^{-i\lambda})$ for a range of $\lambda \in [-\pi, \pi]$ in a Riemannian mesh, and the prospect of doing hundreds of inversions of $N \times N$ complex matrices provides little encouragement to our objectives. The key insight that makes the algorithm successful is an explicit analytical formula for the matrix polynomial inverses $\phi(z)^{-1}$ and $\Phi(z)^{-1}$ in terms of the determinant and adjoint polynomials, whose coefficients can be computed recursively—and then any desired value of $z = e^{-i\lambda}$ may be substituted. This approach can yield dramatic improvements.

2. Algorithmic Details

The spectral density for the VARMA process is defined as

$$F(\lambda) = \Phi(e^{-is\lambda})^{-1}\,\phi(e^{-i\lambda})^{-1}\,\theta(e^{-i\lambda})\,\Theta(e^{-is\lambda})$$
$$\Sigma\,\Theta(e^{is\lambda})'\,\theta(e^{i\lambda})'\,\phi(e^{i\lambda})^{-\prime}\,\Phi(e^{is\lambda})^{-\prime}, \qquad (1)$$

CONTACT Tucker S. McElroy tucker.s.mcelroy@census.gov U.S. Census Bureau, Research and Methodology Directorate, 4600 Silver Hill Road, Washington, DC 20233-0002.

where $-\prime$ is a shorthand for inverse transpose. Then it is known—see Brockwell, Davis, and Fienberg (1991)—that

$$\Gamma_h = \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{ih\lambda} F(\lambda)\, d\lambda. \qquad (2)$$

Therefore, we can compute a series of autocovariances for $0 \leq h \leq H$ by looping over some discretization of $\lambda \in [-\pi, \pi]$ (and looping over h), and computing $\Phi(e^{-is\lambda})^{-1}$ and $\phi(e^{-i\lambda})^{-1}$. However, we wish to avoid computing these inverse matrices for each choice of λ, and we instead compute the adjoint and determinant polynomials. These are defined as follows (see Vu 2008 for background): if $A(z)$ is some matrix polynomial of form $A(z) = \sum_{j=0}^{p} A_j z^j$, then

$$A(z)^{-1} = a(z)^{-1} A(z)^{\sharp}, \qquad (3)$$

where $a(z) = \det A(z)$ and $A(z)^{\sharp}$ is the matrix adjoint polynomial. The determinant polynomial is scalar, of degree pN; the adjoint is a matrix polynomial of degree $p(N-1)$. Both can be computed directly from the coefficients of $A(z)$, so long as the first coefficient matrix A_0 is invertible. (This will always be the case in our applications, because VAR polynomials have I_N for their first coefficient.) Set $B(z) = A_0{}^{-1} A(z)$, so that $B_j = A_0^{-1} A_j$ for $1 \leq j \leq p$. Then $a(z) = \det A_0 \det B(z)$, and $B(z)$ has first coefficient given by I_N; hence, the roots of $\det B(z)$ equal the eigenvalues of the companion matrix of $B(z)$. This companion matrix has the form

$$\begin{bmatrix} -B_1 & -B_2 & \ldots & -B_p \\ I_N & 0 & \ldots & 0 \\ 0 & I_N & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \ldots & I_N & 0 \end{bmatrix} = \begin{bmatrix} -A_0^{-1} A_1 & -A_0^{-1} A_2 & \ldots & -A_0^{-1} A_p \\ I_N & 0 & \ldots & 0 \\ 0 & I_N & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \ldots & I_N & 0 \end{bmatrix}.$$

The dimension is $pN \times pN$, and we must compute the eigenvalues ζ_k $(1 \leq k \leq pN)$, which are the reciprocals of the roots of $\det B(z)$. (Note that s is not involved in this matrix dimension, as we further emphasize below.) Then $\det B(z) = \prod_{k=1}^{pN} (1 - \zeta_k z)$, which determines $a(z)$. Next, from Equation (3) we obtain for $0 \leq k \leq pN$

$$a_k I_N = \sum_{\ell=0}^{p} A_\ell \widetilde{A}_{k-\ell},$$

where $A(z)^{\sharp} = \sum_{\ell=0}^{p(N-1)} \widetilde{A}_\ell$. This relation can be used to recursively compute the adjoint polynomial coefficients:

$$\widetilde{A}_0 = a_0 A_0^{-1}$$
$$\widetilde{A}_\ell = A_0^{-1} \left(a_\ell I_N - \sum_{k=1}^{p \wedge \ell} A_k \widetilde{A}_{\ell-k} \right).$$

Now we simply obtain $\phi(e^{-i\lambda})^{-1}$ by plugging $z = e^{-i\lambda}$ into Equation (3). Similarly, the seasonal VAR polynomial can be inverted; note that $\Phi(z^s)$ is typically viewed as a degree Ps polynomial in z, but for purposes of computing the adjoint and determinant we instead view it as a degree P polynomial in z^s. This is fine, because we only evaluate the inverse at $z^s = e^{-is\lambda}$. As a result, the number of eigenvalues to be computed is only determined by p, P, and N, so that s has no impact whatsoever.

The method is completed by plugging into Equation (1) and discretizing the integral. In our R code we avoid looping over λ by using the usual R gymnastics, but this is unnecessary in the Rcpp version, where we also utilize Simpson's rule in the discretization.

Algorithm.

1. Compute determinant and adjoint polynomials for $\Phi(z)$ and $\phi(z)$.
2. For each λ in a discretization of $[-\pi, \pi]$, compute $\Phi(e^{-is\lambda})^{-1}$ and $\phi(e^{-is\lambda})^{-1}$ from the determinant and adjoint using (3).
3. Compute $F(\lambda)$ for each such λ via (1).
4. For any desired lag h compute Γ_h via the discretization of (2).

3. Numerical Illustration

The github repo https://github.com/tuckermcelroy/VarmaAcf has the four main functions described above: VARMAauto.r is the R implementation of the time domain algorithm, which is described in detail in McElroy (2017). VARMAauto.cpp is the Rcpp version. autoVARMA.r is the R implementation of this paper's frequency domain algorithm, and the Rcpp version is autoVARMA.cpp. A helper function var2.pre2par.r is used to generate the stable VAR polynomial of our example. (Another random example can be generated via setting `psi <- rnorm(p*N*N)`.) With the settings $P = 6$, $N = 7$, and $s = 52$, VARMAauto.r takes 904.32 sec, and VARMAauto.cpp takes 859.44 sec (not much improvement, because our R implementation involves very few loops). In contrast, autoVARMA.r (with a mesh size of 2000, set so as to obtain relative percent error of about 1%) drops to 8.15 sec, and autoVARMA.cpp takes 1.33 sec.

4. Discussion

An essential ingredient to the success of this approach is the factorization of the VAR polynomial into regular and seasonal portions $\phi(z)$ and $\Phi(z^s)$; the seasonal polynomial is of high degree in z, but is of low degree in z^s, and the proposed algorithm takes advantage of this property. Similarly, in situations where the VAR polynomial can be formulated as the product of several factors (each of which is a stable matrix polynomial), we can find the inverse by computing the determinant and adjoint for each factor. In such a scenario, this note's frequency domain approach could also be useful in computing autocovariances. Finally, we remark that the ideas here can be extended to other types of frequency domain calculations, such as the coefficients of a multivariate filter (where the frequency response function involves inverse matrix polynomials) or the inverse autocovariances (in which case we must find the inverse of the VMA polynomial) of a stationary multivariate process.

Acknowledgments

This report is released to inform interested parties of research and to encourage discussion. The views expressed on statistical issues are those of the author and not those of the U.S. Census Bureau.

References

Brockwell, P. J., and Davis, R. A. (1991), *Time Series: Theory and Methods.* Springer Science & Business Media, New York: Springer. [302]

Lütkepohl, H. (2005), *New Introduction to Multiple Time Series Analysis.* Springer Science & Business Media, Berlin: Springer. [301]

McElroy, T. (2017), "Computation of Vector ARMA Autocovariances," *Statistics & Probability Letters*, 124, 92–96. [301,302]

Roy, A., McElroy, T. S., and Linton, P. (2019), "Constrained Estimation of Causal Invertible VARMA," *Statistica Sinica*, 29, 455–478. [301]

Vu, K. M. (2008), "An Extension of the Faddeev's Algorithms," in *2008 IEEE International Conference on Control Applications*, pp. 150–155, IEEE. [302]