Estimating the ARMA Model

John A. Dodson

January 8, 2022

Model

The autoregressive moving average model is useful for extracting latent i.i.d. innovations from a stationary timeseries exhibiting serial autocorrelation.

Let the timeseries be $(X_t)_{t=1,2,\ldots,T}$. The ARMA(1,1) specification is a linear recursion with one autoregressive lag and one moving average lag:

$$X_t = c + \phi X_{t-1} + \theta\varepsilon_{t-1} + \varepsilon_t \quad \text{with i.i.d. } \varepsilon_{1,2,\ldots,T} \sim \mathcal{N}\left(0, \sigma^2\right) \tag{1}$$

initialized with $\varepsilon_0 = 0$ and $X_0 = \frac{c}{1-\phi}$.

Unconditional expectation informs the choice for the initialization. $\mathrm{E}\,\varepsilon_t = 0$ and

$$\begin{aligned} \mathrm{E}\,X_t &= c + \phi\,\mathrm{E}\,X_{t-1} \\ &= c + \phi\,\mathrm{E}\,X_t \\ \Rightarrow \quad (1-\phi)\,\mathrm{E}\,X_t &= c \\ \mathrm{E}\,X_t &= \frac{c}{1-\phi} \end{aligned} \tag{2}$$

To evaluate the (unconditional) variance, introduce the lag operator to re-express the model in terms of the residuals.

$$\begin{aligned} X_t &= c + \phi L X_t + (1+\theta L)\varepsilon_t \\ \Rightarrow \quad (1-L\phi)X_t &= c + (1+\theta L)\varepsilon_t \\ X_t &= \frac{c}{1-\phi} + (1+\phi L + \phi^2 L^2 + \cdots)(1+\theta L)\varepsilon_t \\ &= \frac{c}{1-\phi} + \varepsilon_t + (\phi+\theta)\,\varepsilon_{t-1} + (\phi+\theta)\phi\,\varepsilon_{t-2} + (\phi+\theta)\phi^2\,\varepsilon_{t-3} + \cdots \end{aligned} \tag{3}$$

So, since the residuals are independent,

$$\begin{aligned} \mathrm{var}\,X_t &= \left(1 + (\phi+\theta)^2 + (\phi+\theta)^2\phi^2 + (\phi+\theta)^2\phi^4 + \cdots\right)\sigma^2 \\ &= \left(1 + \frac{(\phi+\theta)^2}{1-\phi^2}\right)\sigma^2 \end{aligned} \tag{4}$$

More about the statistical properties of the model are in the appendix.

Estimation

Since the probability density of the t-th residual is

$$\lim_{\Delta \to 0} \frac{\mathrm{P}\left\{x < \varepsilon_t < x + \Delta\right\}}{\Delta} = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$$

the negative log-likelihood associated with a single observation is

$$\hat{H}_t\left(c, \phi, \theta, \sigma^2\right) = \tfrac{1}{2} \log\left(2\pi\sigma^2\right) + \frac{\hat{\varepsilon}_t^2}{2\sigma^2} \tag{5}$$

where $\hat{\varepsilon}_t$ is defined recursively:

$$\begin{aligned} \hat{\varepsilon}_t &= X_t - c - \phi\, X_{t-1} - \theta\, \hat{\varepsilon}_{t-1} \quad \text{for} \quad t = 2, 3, \ldots, T \\ \hat{\varepsilon}_1 &= X_1 - \frac{c}{1-\phi} \end{aligned} \tag{6}$$

Since the residuals are independent, the negative log-likelihood of the whole sample is just the sum

$$\sum_{t=1}^{T} \hat{H}_t\left(c, \phi, \theta, \sigma^2\right)$$

Recognizing the connection between maximum likelihood and minimum posterior entropy, define

$$\bar{H} = \frac{1}{T} \sum_{t=1}^{T} \hat{H}_t \approx \mathrm{E}\, H_t$$

in terms of which the maximum likelihood estimate of the parameters solves the optimization problem

$$\hat{u} = \arg \min_{c, \phi, \theta, \sigma^2} \bar{H}\left(c, \phi, \theta, \sigma^2\right) \tag{7}$$

Newton's method presents an efficient approach to numerical problems of this sort—especially so in the log-likelihood setting since we can avoid evaluating the hessian explicitly by recalling that

$$\frac{\partial^2 \mathrm{E}\, H_t}{\partial u\, \partial u'} = \operatorname{cov} \frac{\partial H_t}{\partial u'}$$

for $u = \left(c, \phi, \theta, \sigma^2\right)'$ or any equivalent parameterization. So we only need to evaluate the first partials of the likelihood. We can estimate the curvature from their sample covariance.

Partials

The ARMA parameters are implicit in $\hat{H}_t$. They come in through $\hat{\varepsilon}_t$. To evaluate the log-likelihood partials, start with the explicit partials σ^2 and $\hat{\varepsilon}_t$:

$$\frac{\partial \hat{H}_t}{\partial \sigma^2} = \left(1 - \frac{\hat{\varepsilon}_t^2}{\sigma^2}\right) \frac{1}{2\sigma^2} \qquad \frac{\partial \hat{H}_t}{\partial \hat{\varepsilon}_t} = \frac{\hat{\varepsilon}_t}{\sigma^2}$$

hence

$$\frac{\partial \hat{H}_t}{\partial c} = \frac{\hat{\varepsilon}_t}{\sigma^2}\frac{\partial \hat{\varepsilon}_t}{\partial c}$$
$$\frac{\partial \hat{H}_t}{\partial \phi} = \frac{\hat{\varepsilon}_t}{\sigma^2}\frac{\partial \hat{\varepsilon}_t}{\partial \phi}$$
$$\frac{\partial \hat{H}_t}{\partial \theta} = \frac{\hat{\varepsilon}_t}{\sigma^2}\frac{\partial \hat{\varepsilon}_t}{\partial \theta}$$

and the partials of $\hat{\varepsilon}_t$ involve linear recursions similar to the linear recursion of $\hat{\varepsilon}_t$

$$\frac{\partial \hat{\varepsilon}_1}{\partial c} = -\frac{1}{1-\phi} \quad , \qquad \frac{\partial \hat{\varepsilon}_t}{\partial c} = -1 - \theta\frac{\partial \hat{\varepsilon}_{t-1}}{\partial c} \quad \text{for} \quad t = 2, \ldots, T$$
$$\frac{\partial \hat{\varepsilon}_1}{\partial \phi} = -\frac{c}{(1-\phi)^2} \quad , \qquad \frac{\partial \hat{\varepsilon}_t}{\partial \phi} = -X_{t-1} - \theta\frac{\partial \hat{\varepsilon}_{t-1}}{\partial \phi} \quad \text{for} \quad t = 2, \ldots, T$$
$$\frac{\partial \hat{\varepsilon}_1}{\partial \theta} = 0 \quad , \qquad \frac{\partial \hat{\varepsilon}_t}{\partial \theta} = -\hat{\varepsilon}_{t-1} - \theta\frac{\partial \hat{\varepsilon}_{t-1}}{\partial \theta} \quad \text{for} \quad t = 2, \ldots, T$$

This can all be collected into a function to produce the $T \times 4$ quantities that make up

$$\left\{ \left.\frac{\partial \hat{H}_t}{\partial u}\right|_{u_0} \right\}_{t=1,2,\ldots,T}$$

for some candidate values of the parameters u_0, of which we can calculate the sample mean and covariance

$$\begin{aligned} \hat{\mu}_0 &= \frac{1}{T}\sum_{t=1}^{T} \left.\frac{\partial \hat{H}_t}{\partial u'}\right|_{u_0} \\ \hat{\Sigma}_0 &= \frac{1}{T}\sum_{t=1}^{T} \left.\frac{\partial \hat{H}_t}{\partial u'}\right|_{u_0} \left.\frac{\partial \hat{H}_t}{\partial u}\right|_{u_0} \end{aligned} \tag{8}$$

The Newton scheme is

$$u_{i+1} = u_i - \gamma_i \hat{\Sigma}_i^{-1}\hat{\mu}_i \tag{9}$$

for some sequence of step sizes $\gamma_i \in (0, 1]$ each small enough so that $\bar{H}(u_{i+1})$ progressively decreases.

If u_0 is close enough to the solution, this will converge to the maximum likelihood estimate $\hat{u} = u_\infty$ at a quadratic rate, whereby $\hat{\mu}_\infty = 0$ and $\hat{\Sigma}_\infty$ is the asymptotic ($T \to \infty$) Fisher information.

Targeting

In economic applications (for example), it is conventional to incorporate sample moments into the estimation of less easily observed parameters. This is an approximation, of course, but it can be effective. In this case, we can easily evaluate the sample mean and sample variance of $\{X_t\}_{t=1,\ldots,T}$, and if we restrict the search to the parameter subspace defined by these two quantities, the dimension of the problem reduces from four to two.

Fixing the mean and variance at M and V, the negative log-likelihood of an observation becomes

$$\hat{H}'_t(\phi,\theta) = \tfrac{1}{2}\log(2\pi V) - \tfrac{1}{2}\log\left(1+\frac{(\phi+\theta)^2}{1-\phi^2}\right) + \frac{\hat{\varepsilon}'^{\,2}_t}{2V}\left(1+\frac{(\phi+\theta)^2}{1-\phi^2}\right)$$

in terms of the recursion

$$\begin{aligned}\hat{\varepsilon}'_t &= X_t - M - \phi\,(X_{t-1}-M) - \theta\,\hat{\varepsilon}'_{t-1}\\ \hat{\varepsilon}'_1 &= X_1 - M\end{aligned}$$

To simplify the expression of the log-likelihood in this form, introduce the function

$$\gamma(\phi,\theta) = 1+\frac{(\phi+\theta)^2}{1-\phi^2}$$

with partials

$$\begin{aligned}\frac{\partial\gamma}{\partial\phi} &= 2\frac{(\phi+\theta)\,(1+\phi\,\theta)}{(1-\phi^2)^2}\\ \frac{\partial\gamma}{\partial\theta} &= 2\frac{\phi+\theta}{1-\phi^2}\end{aligned}$$

In these terms, the partials of $\hat{H}'_t$ are

$$\begin{aligned}\frac{\partial\hat{H}'_t}{\partial\phi} &= \left(\frac{\hat{\varepsilon}'^{\,2}_t}{V/\gamma}-1\right)\frac{(\phi+\theta)\,(1+\phi\,\theta)}{(1+2\phi\,\theta+\theta^2)\,(1-\phi^2)} + \frac{\hat{\varepsilon}'_t}{V/\gamma}\frac{\partial\hat{\varepsilon}'_t}{\partial\phi}\\ \frac{\partial\hat{H}'_t}{\partial\theta} &= \left(\frac{\hat{\varepsilon}'^{\,2}_t}{V/\gamma}-1\right)\frac{\phi+\theta}{1+2\phi\,\theta+\theta^2} + \frac{\hat{\varepsilon}'_t}{V/\gamma}\frac{\partial\hat{\varepsilon}'_t}{\partial\theta}\end{aligned}$$

and

$$\begin{aligned}\frac{\partial\hat{\varepsilon}'_t}{\partial\phi} &= M - X_t - \theta\frac{\partial\hat{\varepsilon}'_{t-1}}{\partial\phi} \quad , \quad \frac{\partial\hat{\varepsilon}'_1}{\partial\phi} = 0\\ \frac{\partial\hat{\varepsilon}'_t}{\partial\theta} &= -\hat{\varepsilon}'_{t-1} - \theta\frac{\partial\hat{\varepsilon}'_{t-1}}{\partial\theta} \quad , \quad \frac{\partial\hat{\varepsilon}'_1}{\partial\theta} = 0\end{aligned}$$

Note that, to the extent that γ is constant close to the solution, minimizing $\bar{H}'$ is essentially minimizing the sum of squared residuals[1]. Presumably this is a motivation for the heuristic of using linear regression to identify ARMA parameters that one sometimes encounters in the literature. Ones hopes those authors are aware of this limitation, which in particular depends on $\phi+\theta$ being close to zero.

[1]The implementation in the R statistics domain-specific language uses a mis-named 'exact maximum likelihood' algorithm which is also based on minimizing the sum of squared residuals.

References

[1] Larry Armijo. Minimization of functions having Lipschitz continuous first partial derivatives. *Pacific Journal of Mathematics*, 16(1):1–3, January 1966.

[2] Ernst K. Berndt, Bronwyn H. Hall, Robert E. Hall, and Jerry A. Hausman. Estimation and inference in nonlinear structural models. *Annals of Economic and Social Measurement*, 3(4):653–665, October 1974.

Appendix A: Serial Autocorrelation

From (3), the expression of the process in terms of the latent i.i.d. residuals, it is straight-forward to evaluate

$$\operatorname{cov}(X_t, X_{t-h}) = \phi^{h-1} \frac{(\phi+\theta)(1+\phi\theta)}{1-\phi^2}\sigma^2$$

for $h = 1, 2, \cdots$.

Dividing through by the variance we get the autocorrelation, which has an elegant form:

$$\rho_h = \operatorname{cor}(X_t, X_{t-h}) = \frac{\phi^{h-1}}{\frac{1}{\phi+\theta} + \frac{1}{\phi+1/\theta}}$$

This result has several consequences. First, since correlation is only meaningful if it is between -1 and $+1$, it is clear that $|\phi| \leq 1$. Second, it seems that θ and $1/\theta$ are interchangeable in the expression for correlation. In fact, the expression for the variance can also be written as

$$\begin{aligned}\operatorname{var} X_t &= \left(1 + \frac{(\phi+\theta)^2}{1-\phi^2}\right)\sigma^2 \\ &= \left(1 + \frac{(\phi+1/\theta)^2}{1-\phi^2}\right)\theta^2\sigma^2\end{aligned}$$

so replacing θ by its inverse can be compensated for by scaling the residual variance by θ^2 (and reversing the order and possibly signs of the i.i.d. residual variates). So, without loss of generality, we can assume $|\theta| \leq 1$. Third, the supported autocorrelation has a particular functional form,

$$\begin{aligned}\rho_h &= \phi\,\rho_{h-1} \quad , \quad h = 2, 3, \ldots \\ \rho_1 &= \frac{1}{\frac{1}{\phi+\theta} + \frac{1}{\phi+1/\theta}}\end{aligned}$$

Hence ϕ and θ are determined by the first two autocorrelations:

$$\phi = \begin{cases} \frac{\rho_2}{\rho_1} & \rho_1 \neq 0 \\ 0 & \rho_1 = 0 \end{cases}$$

$$\theta = \begin{cases} \zeta^{-1} - \operatorname{sgn}(\zeta)\sqrt{\zeta^{-2}-1} & \zeta \neq 0 \\ 0 & \zeta = 0 \end{cases}$$

$$\text{where} \quad \zeta = \frac{2\rho_1\left(\rho_1^2 - \rho_2\right)}{\rho_1^2(1-2\rho_2) + \rho_2^2}$$

for any $-1 \leq \rho_1 \leq 1$ and $|\rho_1|\,(2|\rho_1| - 1) \leq \rho_2 \leq |\rho_1|$.

Appendix B: Sample Julia[2] implementation

```
module ARMA

"residual timeseries for the ARMA model for X for parameters vector [c,ϕ,θ]"
function arma(X::Vector,params::Vector)
    n = length(X)
    c,ϕ,θ = params
    ε = fill(NaN,n)
    if abs(ϕ) < 1 && abs(θ) < 1
        ε[1] = X[1]-c/(1-ϕ)
        for i = 2:n
            ε[i] = X[i]-c-ϕ*X[i-1]-θ*ε[i-1]
        end
    end
    return ε
end

"gradient of ARMA residual wrt parameters [c,ϕ,θ] in the form of a timeseries of vectors"
function arma_grad(X::Vector,params::Vector)
    n = length(X)
    c,ϕ,θ = params
    ε = arma(X,params)
    ∇ = fill(fill(NaN,3),n)
    if abs(ϕ) < 1 && abs(θ) < 1
        ∇[1] = [-1/(1-ϕ), -c/(1-ϕ)^2, 0]
        for i = 2:n
            ∇[i] = [ -1-θ*∇[i-1][1], -X[i-1]-θ*∇[i-1][2], -ε[i-1]-θ*∇[i-1][3] ]
        end
    end
    return ∇
end

"""timeseries of the negative quasi log-likelihood of the ARMA fit of X
with parameters [σ²,c,ϕ,θ] in the form of a vector"""
function h(X::Vector,u::Vector)
    σ² = u[1]
    params = u[2:end]
    if σ² ≤ 0
        return fill(NaN,length(X))
    end
    ε = arma(X,params)
    return (log(2π*σ²).+ε.^2/σ²)/2
end
```

[2]https://julialang.org

```
"""gradient of the negative quasi log-likelihood of the ARMA fit
in the form of a timeseries of vectors"""
function h_grad(X::Vector,u::Vector)
    σ² = u[1]
    params = u[2:end]
    if σ² ≤ 0
        return fill(fill(NaN,length(u)),length(X))
    end
    ε = arma(X,params)
    return vcat.((1 .-ε.^2/σ²)/(2σ²), ε/σ².*arma_grad(X,params))
end

"Newton's method minimizer"
function newt(func::Function,grad::Function,hess::Function,u₀::Vector
                    ;maxiter=100,tol=1.E-14,δ=1.E-4)
    u₁ = u₀
    h₁ = func(u₁)
    if isnan(h₁)
        throw(DomainError(u₀,"invalid initial value"))
    end
    N = maxiter
    k = 0
    while N > 0
        u₀ = u₁
        h₀ = h₁
        grad₀ = grad(u₀)
        hess₀ = hess(u₀)
        k = 0
        while N > 0 && ( k == 0 || isnan(h₁) || h₁ > h₀+δ*transpose(u₁-u₀)*grad₀ )
            u₁ = u₀-hess₀\grad₀/2^k
            h₁ = func(u₁)
            k += 1
            N -= 1
        end
        if abs(h₁-h₀) < tol
            return u₁
        end
    end
    throw(ErrorException("convergence criterion not met at u = $u₁"))
end

"""BHHH scheme for the maximum likelihood estimator for the ARMA model
with parameters [σ²,c,ϕ,θ]. Inferred residuals are i.i.d. normal with variance σ²"""
function arma_mle(X::Vector)
```

```
    n = length(X)
    func = u -> sum(h(X,u))/n
    grad = u -> sum(h_grad(X,u))/n
    hess = u -> sum([∇*transpose(∇) for ∇ in h_grad(X,u)])/n
    M = sum(X)/n
    V = sum((X.-M).^2)/n
    ϕ₀,θ₀ = 0.,0.
    u₀ = [V/(1+(θ₀+ϕ₀)^2/(1-ϕ₀^2)),M*(1-ϕ₀),ϕ₀,θ₀]
    return newt(func,grad,hess,u₀)
end

"log-likelihood for the ARMA fit to timeseries X"
function arma_log_likelihood(X::Vector,u::Vector)
    return -sum(h(X,u))
end

"Fisher information matrix for parameters [σ²,c,ϕ,θ] with respect to timeseries sample X"
function arma_fisher_information(X::Vector,u::Vector)
    return sum([∇*transpose(∇) for ∇ in h_grad(X,u)])/length(X)
end


export arma, arma_mle, arma_log_likelihood, arma_fisher_information

end # module
```