Estimating MA and ARMA Parameters

Jared Fisher

Lecture 10b

Recap: Estimating AR(p) Parameters

Estimating AR(p) Parameters

- Yule-Walker (Method of Moments): plug in sample ACVF values as the "true" values in Yule-Walker equations
- Least squares: choose parameter values that minimize the squared errors $X_t - \hat{X}_t$
- Maximum-likelihood: assume a distribution on W_t (e.g. Guassian noise), then choose parameter values that maximize that likelihood

Parameter Estimation in ARMA

Method of Moments or Yule-Walker Method

The process, in principle, of solving some subset of equations for the unknown parameters $\theta_1, \ldots, \theta_q, \phi_1, \ldots, \phi_p$ and σ_W^2 (and μ is estimated by the sample mean), by plugging in the sample acvf $\hat{\gamma}(k)$ as an estimate for the true acvf $\gamma(k)$, such as

$$\hat{\gamma}(k) - \phi_1\hat{\gamma}(k-1) - \cdots - \phi_p\hat{\gamma}(k-p) = (\psi_0\theta_k + \psi_1\theta_{k+1} + \cdots + \psi_{q-k}\theta_q)\sigma_W^2$$

for $0 \leq k \leq q$ and

$$\hat{\gamma}(k) - \phi_1\hat{\gamma}(k-1) - \cdots - \phi_p\hat{\gamma}(k-p) = 0 \text{ for } k > q$$

can in principle be applied for ARMA(p,q) models, as well. Note that ψ_j above are functions of $\theta_1, \ldots, \theta_q$ and $\phi_1, \ldots, \phi_p$.

Example: MA(1)

- For an invertible MA(1) model $X_t = W_t + \theta W_t$,

$$\gamma_X(0) = \sigma_W^2(1+\theta^2) \quad \text{and } \gamma_X(1) = \sigma_W^2\theta.$$

- Thus, with the method of moments one would estimate θ by solving

$$r_1 = \hat{\rho}(1) = \frac{\hat{\gamma}(1)}{\hat{\gamma}(0)} = \frac{\hat{\theta}}{(1+\hat{\theta}^2)}$$

- (two solutions exists, so we would pick the invertible one).
- The problem with this estimator is, that the above equation only has a solution when $|r_1| \leq 1/2$. Although $|\rho(1)| \leq 1/2$, because $\hat{\rho}(1)$ is just an estimate, it does not always hold true that $|r_1| \leq 1/2$.
- R code: try out some simulations in R with $|\rho(1)| \approx 1/2$ and check that $|r_1| > 1/2$ quite often.

Problem

In general, the method of moments for ARMA(p,q) models, has two major problems:

1. It is cumbersome (unless we are in the pure AR case): Solutions might not always exist to these equations (like in this last example).The parameters are estimated in an arbitrary fashion when these equations do not have a solution.
2. The estimators obtained are *inefficient*. Other techniques give much better estimates (smaller standard errors).

- Because of these problems, no one uses method of moments for estimating the parameters of a general ARMA model. R does not even have a function for doing this. Note, however, that both of these problems disappear for the case of the pure AR model.

R Code

- We'll now examine the arima() function in R.
- First, a clarification. The arima() function uses the non-zero mean AR(1) model if you don't take a difference:

$$(X_t - \mu) = \phi(X_{t-1} - \mu) + W_t$$

or

$$X_t = \mu + \phi(X_{t-1} - \mu) + W_t$$

Conditional least squares

We'll start by looking at two examples.

Example: MA(1)'s CLS

- For an MA(1) model we have $X_t - \mu = W_t + \theta W_{t-1}$. We want to fit this model to data $x_1, \ldots, x_n$.
- If the data were indeed generated from this model, then

$$
\begin{aligned}
W_1 &= x_1 - \mu - \theta W_0 \\
W_2 &= x_2 - \mu - \theta W_1 \\
&\vdots \\
W_n &= x_n - \mu - \theta W_{n-1}
\end{aligned}
$$

- If we set $W_0 = E(W_0) = 0$, then we can
  - recursively calculate $W_1, \ldots, W_n$ as a function of μ and θ
  - compute the sum of squares $\sum_{i=1}^{n} W_i^2$
  - choose μ and θ such that they minimize this sum of squares
- This is also called conditional least squares because this minimization is obtained when one tries to maximize the likelihood conditioning on $W_0 = 0$.

Example: ARMA(1,1)'s CLS

- Here the model is $X_t - \mu - \phi(X_{t-1} - \mu) = W_t + \theta W_{t-1}$.
- Here it is convenient to set W_1 to be zero. Then we can write

$$
\begin{aligned}
W_2 &= x_2 - \mu - \phi(x_1 - \mu) \\
W_3 &= x_3 - \mu - \phi(x_2 - \mu) - \theta W_2 \\
&\vdots \\
W_n &= x_n - \mu - \phi(x_{n-1} - \mu) - \theta W_{n-1}
\end{aligned}
$$

- Then the sum of squares $\sum_{i=2}^{n} W_i^2$ is a function of θ, ϕ, and μ, which can be minimized.

Definition: Conditional least squares for ARMA(p,q)

Given some data $x_1, \ldots, x_n$ and $p, q \in N$, define a function $S_c(\mu, \phi_1, \ldots, \phi_p, \theta_1, \ldots, \theta_q)$ as follows:

1. Set $W_t = 0$ for all $t \leq p$.
2. For $t = p + 1, \ldots, n$, recursively calculate:

$$W_t = X_t - \mu - \phi_1(X_{t-1} - \mu) - \cdots - \phi_p(X_{t-p} - \mu) - \theta_1 W_{t-1} - \cdots - \theta_q W_{t-q}$$

3. Let $S_c(\mu, \phi_1, \ldots, \phi_p, \theta_1, \ldots, \theta_q) = \sum_{t=p+1}^{n} W_t^2$.

Then the conditional last squares estimator $\hat{\mu}, \hat{\phi}_1, \ldots, \hat{\phi}_p, \hat{\theta}_1, \ldots, \hat{\theta}_q$ is defined by minimizing the conditional sum of squares

$$S_c(\hat{\mu}, \hat{\phi}_1, \ldots, \hat{\phi}_p, \hat{\theta}_1, \ldots, \hat{\theta}_q) = \min_{\mu, \phi, \theta} S_c(\mu, \phi_1, \ldots, \phi_p, \theta_1, \ldots, \theta_q)$$

Comments on Definition

- This is equivalent to writing the likelihood conditioning on $X_1, \ldots, X_p$ and $W_t = 0$ for $t \leq p$.
- If $q = 0$ (AR models), minimizing the sum of squares is equivalent to linear regression and no iterative technique is needed.
- If $q > 0$, the problem becomes nonlinear regression and numerical optimization routines need to be used.
- In R, this method is performed by calling the function arima() with the method argument set to *CSS* (CSS stands for conditional sum of squares).
- As before, we can estimate the noise variance via

$$\hat{\sigma}_W^2 = \frac{S_c(\hat{\mu}, \hat{\phi}, \hat{\theta})}{n - p}.$$

Maximum Likelihood

- Assume that errors $\{W_t\}$ are Gaussian.
- Write down the likelihood of the observed data $x_1, x_2, \ldots, x_n$ in terms of the unknown parameter values $\mu, \theta_1, \ldots, \theta_q$, $\phi_1, \ldots, \phi_p$ and σ_W^2.
- Maximize over these unknown parameter values.
- R: use the function arima() with the method argument set to *ML*
- ML stands or Maximum Likelihood. R uses an optimization routine to maximize the likelihood. This routine is iterative and needs suitable initial values of the parameters to start.
- You can also set method equal to *CSS-ML*, where R selects the starting values by CSS.

Asymptotic Distribution of Estimators

- See Property 3.10 in TSA4e
- Yields the SE's on coefficients from arima()