universität
innsbruck

Time Series Analysis

Parameter Estimation

Parameter Estimation

Contents

Contents

Contents

Goal: Estimation of the unknown parameters in an ARIMA model for an observed time series $Y_1, \ldots, Y_n$ *given* a specification of the orders p, d, and q.

Note: Given d, an ARMA(p, q) model has to be estimated for the d-th difference of the Y_t.

Hence: In the following, we assume $d = 0$, without loss of generality. Otherwise Y_t would just be the d-th difference of the original series.

Parameter Estimation

The Method of Moments

The method of moments

Idea:

- Equate empirical sample moments to corresponding theoretical moments (i.e., functions of the unknown parameters).
- Solve resulting equations for unknown parameters.

Example: Estimate mean of a stationary process by the sample mean.

Properties:

- Often easy to implement.
- Typically not efficient.
- Can be employed as starting values to other estimation methods with improved properties.

The method of moments: AR(p)

Example: AR(1). The theoretical autocorrelation at lag 1 for AR(1) process with AR coefficient ϕ is

$$\varrho_1 = \phi$$

Thus, plugging in the sample autocorrelation r_1 at lag 1 yields the method of moments (MM) estimator

$$\hat{\phi} = r_1$$

Example: AR(2). Yule-Walker equations for the theoretical moments

$$\varrho_1 = \phi_1 + \varrho_1\phi_2 \quad \text{and} \quad \varrho_2 = \varrho_1\phi_1 + \phi_2$$

Hence, the MM estimators fulfill the analogous equations for the empirical moments

$$r_1 = \hat{\phi}_1 + r_1\hat{\phi}_2 \quad \text{and} \quad r_2 = r_1\hat{\phi}_1 + \hat{\phi}_2$$

The method of moments: AR(p)

Solving for $\hat{\phi}_1$ and $\hat{\phi}_2$ yields

$$\hat{\phi}_1 = \frac{r_1(1 - r_2)}{1 - r_1^2} \quad \text{and} \quad \hat{\phi}_2 = \frac{r_2 - r_1^2}{1 - r_1^2}$$

More generally: AR(p). The MM estimates can be employed by solving the empirical counterparts of the Yule-Walker equations, hence also known as *Yule-Walker estimates*.

$$\begin{aligned}
r_1 &= \hat{\phi}_1 + \hat{\phi}_2 r_1 + \hat{\phi}_3 r_2 + \ldots + \hat{\phi}_p r_{p-1} \\
r_2 &= \hat{\phi}_1 r_1 + \hat{\phi}_2 + \hat{\phi}_3 r_1 + \ldots + \hat{\phi}_p r_{p-2} \\
&\vdots \\
r_p &= \hat{\phi}_1 r_{p-1} + \hat{\phi}_2 r_{p-2} + \hat{\phi}_3 r_{p-3} + \ldots + \hat{\phi}_p
\end{aligned}$$

Durbin-Levinson recursion can be used for solving equations. However, the method is subject to substantial round-off errors for solutions close to the boundary of the stationary region.

The method of moments: MA(q)

Example: MA(1). Theoretical autocorrelation at lag 1 is

$$\varrho_1 = -\frac{\theta}{1+\theta^2}$$

Equating $\varrho_1 = r_1$ and solving for θ yields two roots. Provided that $|r_1| < 0.5$:

$$-\frac{1}{2r_1} \pm \sqrt{\frac{1}{4r_1^2} - 1}$$

Product of both roots is always 1 and only one solution satisfies invertibility condition $|\theta| < 1$:

$$\hat{\theta} = \frac{-1 + \sqrt{1 - 4r_1^2}}{2r_1}$$

The method of moments: MA(q)

Remarks:

- For $|r_1| = 0.5$, real solutions exist with $|\theta| = 1$, but are not invertible.
- For $|r_1| > 0.5$, no real solutions exist and thus MM fails. (However, MA(1) specification would also be in doubt.)
- For MA(q), MM estimation quickly gets complicated due to nonlinearity of equations. Also, only one from multiple solutions would be invertible.
- Furthermore, MM estimates for models with MA terms can be shown to be very inefficient.
- Hence, MM rarely used in practice for MA and ARMA models.

The method of moments: ARMA(1, 1)

Example: ARMA(1, 1). The autocorrelation function is given by

$$\varrho_k \;=\; \frac{(1-\theta\phi)(\phi-\theta)}{1-2\theta\phi+\theta^2}\,\phi^{k-1} \qquad \text{for } k \geq 1$$

As $\varrho_2/\varrho_1 = \phi$, estimate

$$\hat{\phi} \;=\; \frac{r_2}{r_1}$$

And then solve

$$r_1 \;=\; \frac{(1-\hat{\theta}\hat{\phi})(\hat{\phi}-\hat{\theta})}{1-2\hat{\theta}\hat{\phi}+\hat{\theta}^2}$$

for $\hat{\theta}$.

Again, the invertible solution of the quadratic equation, if any, is used.

The method of moments: Noise variance

Idea: The process variance $\gamma_0 = \text{Var}(Y_t)$ can always be estimated by the sample variance

$$s^2 = \frac{1}{n-1}\sum_{t=1}^{n}(Y_t - \bar{Y})^2$$

Then, the known relationship between γ_0 and σ_e^2, the ϕs and θs can be solved for σ_e^2.

AR(p):

$$\hat{\sigma}_e^2 = (1 - \hat{\phi}_1 r_1 - \cdots - \hat{\phi}_p r_p)\, s^2$$

Thus, AR(1):

$$\hat{\sigma}_e^2 = (1 - r_1^2)\, s^2$$

The method of moments: Noise variance

MA(q):

$$\hat{\sigma}_e^2 = \frac{s^2}{1 + \hat{\theta}^2 + \cdots + \hat{\theta}_q^2}$$

ARMA(1, 1):

$$\hat{\sigma}_e^2 = \frac{1 - \hat{\phi}^2}{1 - 2\hat{\phi}\hat{\theta} + \hat{\theta}^2}\, s^2$$

The method of moments

In R:

- `ar()` estimates AR(p) models using various methods, including `method = "yule-walker"` (or `"yw"` for short).
- By default, this demeans the series (`demean = TRUE`) and employs automatic selection of p via AIC (`aic = TRUE`).
- MM estimates for MA or ARIMA models are not directly available in the base R packages.

Illustration: Parameter estimation for simulated MA(1), AR(1), and AR(2) series from previous chapters.

The method of moments

Simulate MA(1) with $\theta = -0.9$

```
R> set.seed(1)
R> ma1a <- arima.sim(model = list(ma = 0.9), n = 120)
```

Convenience function for MM estimate $\hat{\theta}$ in MA(1) models:

```
R> ma1fit <- function(x) {
+    r1 <- acf(x, plot = FALSE)$acf[2]
+    if(abs(r1) > 0.5) NA else (-1 + sqrt(1 - 4 * r1^2))/(2 * r1)
+  }
R> ma1fit(ma1a)
[1] -0.8801
```

For one simulated MA(1) with $\theta = 0.9$

```
R> set.seed(1)
R> ma1b <- arima.sim(model = list(ma = -0.9), n = 120)
R> ma1fit(ma1b)
[1] NA
R> acf(ma1b, plot = FALSE)$acf[2]
[1] -0.5053
```

The method of moments

Simulate AR(1) with $\phi = 0.9$

```
R> set.seed(0)
R> ar1a <- arima.sim(model = list(ar = 0.9), n = 120)
```

Estimate $\hat{\phi}$ can be obtained directly from `acf()` or, equivalently, from `ar()` with `method = "yule-walker"`

```
R> acf(ar1a, plot = FALSE)$acf[2]
[1] 0.8867
R> ar(ar1a, order.max = 1, aic = FALSE, method = "yule-walker")
Call:
ar(x = ar1a, aic = FALSE, order.max = 1, method = "yule-walker")

Coefficients:
    1
0.887

Order selected 1  sigma^2 estimated as  0.855
```

The method of moments

Simulated AR(1) with $\phi = -0.8$

```
R> set.seed(2)
R> ar1b <- arima.sim(model = list(ar = -0.8), n = 120)
R> ar(ar1b, order.max = 1, aic = FALSE, method = "yule-walker")
```

```
Call:
ar(x = ar1b, aic = FALSE, order.max = 1, method = "yule-walker")

Coefficients:
     1
-0.799

Order selected 1  sigma^2 estimated as  1.33
```

The method of moments

Simulated AR(2) with $\phi_1 = 1$, $\phi_2 = -0.6$

```
R> set.seed(1)
R> ar2 <- arima.sim(model = list(ar = c(1, -0.6)), n = 120)
R> ar(ar2, order.max = 2, aic = FALSE, method = "yule-walker")
```

```
Call:
ar(x = ar2, aic = FALSE, order.max = 2, method = "yule-walker")

Coefficients:
     1       2
 1.039  -0.587

Order selected 2  sigma^2 estimated as  0.779
```

Parameter Estimation

Least Squares Estimation

Least squares estimation

Motivation: As MM estimation is often unsatisfactory, especially for MA and ARIMA models, consider least squares estimation. Again, straightforward for AR models.

Also: Incorporate a (potentially nonzero) mean μ in the model.

Idea: For AR(1).

$$\begin{aligned} Y_t - \mu &= \phi(Y_{t-1} - \mu) + e_t \\ e_t &= (Y_t - \mu) - \phi(Y_{t-1} - \mu) \end{aligned}$$

For estimation of ϕ and μ, minimize the squared residuals. Note that these are only observed for $t = 2, \ldots, n$, yielding the so-called *conditional sum of squares* (conditional on Y_1).

$$S_c(\phi, \mu) = \sum_{t=2}^{n} [(Y_t - \mu) - \phi(Y_{t-1} - \mu)]^2$$

Least squares estimation: AR(1)

$$\frac{\partial S_c}{\partial \mu} = \sum_{t=2}^{n} 2[(Y_t - \mu) - \phi(Y_{t-1} - \mu)](-1 + \phi) = 0$$

yields

$$\mu = \frac{1}{(n-1)(1-\phi)} \left[\sum_{t=2}^{n} Y_t - \phi \sum_{t=2}^{n} Y_{t-1} \right]$$

For large n

$$\frac{1}{n-1} \sum_{t=2}^{n} Y_t \approx \frac{1}{n-1} \sum_{t=2}^{n} Y_{t-1} \approx \bar{Y}$$

And thus, independent of the value of ϕ,

$$\hat{\mu} \approx \frac{1}{1-\phi}(\bar{Y} - \phi\bar{Y}) = \bar{Y}$$

Jargon: Except for *end effects*, $\hat{\mu} = \bar{Y}$.

Least squares estimation: AR(1)

$$\frac{\partial S_c(\phi, \bar{Y})}{\partial \phi} = \sum_{t=2}^{n} 2[(Y_t - \bar{Y}) - \phi(Y_{t-1} - \bar{Y})](Y_{t-1} - \bar{Y}) = 0$$

yields

$$\hat{\phi} = \frac{\sum_{t=2}^{n}(Y_t - \bar{Y})(Y_{t-1} - \bar{Y})}{\sum_{t=2}^{n}(Y_{t-1} - \bar{Y})^2}$$

Note: Except for one term in the denominator, $(Y_n - \bar{Y})^2$, $\hat{\phi}$ is identical to the Yule-Walker estimate r_1.

Analogously: AR(p). Yule-Walker estimates are close to OLS estimates.

Least squares estimation: MA(1)

Next: MA(1). Invertible process can be written as

$$\begin{aligned} Y_t &= e_t - \theta e_{t-1} \\ &= -\theta Y_{t-1} - \theta^2 Y_{t-2} - \theta^3 Y_{t-3} - \ldots + e_t \\ e_t &= Y_t + \theta e_{t-1} \\ &= Y_t + \theta Y_{t-1} + \theta^2 Y_{t-2} + \theta^3 Y_{t-3} + \ldots \end{aligned}$$

For computing (and then optimizing) the sum of squares

$$S_c(\theta) = \sum e_t^2$$

the AR(∞) representation is difficult. However, *conditional on* $e_0 = 0$, the e_t can be computed recursively.

$$\begin{aligned} e_1 &= Y_1 \\ e_2 &= Y_2 + \theta e_1 \\ &\vdots \end{aligned}$$

Least squares estimation: MA(1)

Remarks:

- $S_c(\theta)$ is nonlinear in θ.
- Hence, numerical methods are typically required for optimization, e.g., Nelder-Mead or Gauss-Newton etc.
- MA(q) works analogously with $e_0 = e_{-1} = \cdots = e_{-q+1} = 0$.
- In large samples, the influence of the starting values will have only little influence.

Least squares estimation: ARMA(1, 1)

Furthermore: ARMA(1, 1). Employing the same ideas as above

$$\begin{aligned} Y_t &= \phi Y_{t-1} + e_t - \theta e_{t-1} \\ e_t &= Y_t - \phi Y_{t-1} + \theta e_{t-1} \end{aligned}$$

Initialize with $e_1 = 0$ and Y_1 and minimize

$$S_c(\phi, \theta) = \sum_{t=2}^{n} e_t^2$$

Analogously: ARMA(p, q).

Least squares estimation

In R:

- `ar(..., method = "ols")` estimates AR(p) models by conditional least squares. By default, μ is included (`demean = TRUE`).
- `arima(..., method = "CSS")` estimates ARIMA(p, d, q) models by conditional least squares. Similarly, to `ar()` an overall intercept is included by default `include.mean`.
- `dynlm()` from *dynlm* can also be employed for OLS estimation of models like `dynlm(y ~ L(y))` or `dynlm(y ~ L(y, 1:3))`. This uses the OLS regression parametrization with an intercept.

Parameter Estimation

Maximum Likelihood and Unconditional Least Squares

Maximum likelihood

Motivation:

- For series of moderate length or seasonal models, the starting values might have a more pronounced effect.
- Employ not only information in first two moments, but in full likelihood.
- Large-sample theory is established under fairly general conditions for maximum likelihood (ML) estimation.

Question: What is the joint distribution of $Y_1, \ldots, Y_n$?

Recall:

- Likelihood L of observations $Y_1, \ldots, Y_n$ is the value of their joint density, considered as a function of the unknown parameters.
- ML estimators of the parameters are those values for which observing the actual data is most likely.

Maximum likelihood: AR(1)

Example: AR(1). Assume that the innovations e_t are n.i.d.(0, σ_e^2), i.e., independent normally distributed variables with zero mean and constant variance σ_e^2.

The probability density function (PDF) of each e_t is

$$\frac{1}{\sqrt{2\pi\sigma_e^2}} \exp\left(-\frac{e_t^2}{2\sigma_e^2}\right) \qquad \text{for } -\infty < e_t < \infty$$

Due to independence the joint PDF for $e_2, \ldots, e_n$ is

$$\left(\frac{1}{\sqrt{2\pi\sigma_e^2}}\right)^{n-1} \exp\left(-\frac{1}{2\sigma_e^2}\sum_{t=2}^{n} e_t^2\right)$$

Conditioning on $Y_1 = y_1$, $Y_t - \mu = \phi(Y_{t-1} - \mu) + e_t$ for $t = 2, \ldots, n$ is a linear transformation.

Maximum likelihood: AR(1)

Obtain conditional PDF for $Y_2, \ldots, Y_n$ by substitution

$$
\begin{aligned}
f(y_2, \ldots, y_n \mid y_1) &= \left(\frac{1}{\sqrt{2\pi\sigma_e^2}}\right)^{n-1} \cdot \\
&\quad \exp\left\{-\frac{1}{2\sigma_e^2}\sum_{t=2}^{n}[(y_t-\mu)-\phi(y_{t-1}-\mu)]^2\right\}
\end{aligned}
$$

Marginal distribution of Y_1 is $\mathcal{N}(\mu, \gamma_0)$ with $\gamma_0 = \sigma_e^2/(1-\phi^2)$ and thus

$$
\begin{aligned}
f(y_1) &= \frac{1}{\sqrt{2\pi\gamma_0}} \exp\left(-\frac{(y_1-\mu)^2}{2\gamma_0}\right) \\
&= \frac{1}{\sqrt{2\pi\sigma_e^2}} \sqrt{1-\phi^2} \exp\left(-\frac{1}{2\sigma_e^2}(1-\phi^2)(y_1-\mu)^2\right)
\end{aligned}
$$

Maximum likelihood: AR(1)

Thus, specification of likelihood is complete:

$$
\begin{aligned}
L(\phi, \mu, \sigma_e^2) &= f(y_2, \ldots, y_n \mid y_1)\, f(y_1) \\
&= \left(\frac{1}{\sqrt{2\pi\sigma_e^2}}\right)^n \sqrt{1-\phi^2}\ \exp\left(-\frac{1}{2\sigma_e^2} S(\phi, \mu)\right) \\
S(\phi, \mu) &= (1-\phi^2)(y_1-\mu)^2 \ + \ \sum_{t=2}^{n} [(y_t - \mu) - \phi(y_{t-1} - \mu)]^2
\end{aligned}
$$

where $S(\phi, \mu)$ is the *unconditional sum of squares*.

Both for numerical and analytical analysis, the *log-likelihood* $\ell(\phi, \mu, \sigma_e^2) = \log L(\phi, \mu, \sigma_e^2)$ is typically more convenient.

Maximum likelihood: AR(1)

Given ϕ and μ, $\ell(\phi, \mu, \sigma_e^2)$ can be easily maximized analytically wrt. σ_e^2, yielding

$$\hat{\sigma}_e^2 = \frac{S(\hat{\phi}, \hat{\mu})}{n}$$

Finite sample correction with $n - 2$ is often applied to reduce bias.

For estimating ϕ and μ minimize

$$S(\phi, \mu) = S_c(\phi, \mu) + (1 - \phi^2)(Y_1 - \mu)^2$$

where in large sample $S(\phi, \mu) \approx S_c(\phi, \mu)$.

The effect of the last term is more substantial for $|\phi|$ close to 1.

Maximum likelihood: ARMA(p, q)

More generally: Employ the same ideas for general ARMA(p, q) models. However, derivation is much more complex.

In R:

- `arima()` estimates general ARIMA(p, d, q) models.
- ML estimation can be performed using `method = "ML"` or `method = "CSS-ML"` (default).
- `"CSS-ML"` obtains initial estimates via CSS and uses them as input for the numerical optimization of the log-likelihood.
- Estimates for the associated covariance matrix are obtained from the associated Hessian.

Parameter Estimation

Properties of the Estimates

Properties of the estimates

Central limit theorem: Let $\{Y_t\}$ be an ARMA(p, q) process (stationary, invertible, i.e., with all roots of $\phi(B)$ and $\theta(B)$ outside the unit circle and no common roots) and $\{e_t\}$ i.i.d. with zero mean and variance σ_e^2. Then, for the maximum likelihood estimator of the full parameter vector $\beta = (\phi_1, \ldots, \phi_p, \theta_1, \ldots, \theta_q)^\top$

$$\sqrt{n}\,(\hat{\beta} - \beta) \xrightarrow{d} \mathcal{N}(0, V(\beta))$$

Properties of the estimates

Covariance matrix: Given by

$$V(\beta) \;=\; \begin{pmatrix} \mathsf{E}(U_t U_t^\top) & \mathsf{E}(U_t V_t^\top) \\ \mathsf{E}(V_t U_t^\top) & \mathsf{E}(V_t V_t^\top) \end{pmatrix}^{-1}$$

where $U_t = (u_t, u_{t-1}, \ldots, u_{t-p+1})^\top$ and $V_t = (v_t, v_{t-1}, \ldots, v_{t-q+1})^\top$ are AR processes for some white noise process $\{w_t\}$ with $\sigma_w^2 = 1$ and

$$\begin{aligned} w_t &= u_t - \phi_1 u_{t-1} - \ldots - \phi_p u_{t-p} \\ &= v_t - \theta_1 v_{t-1} - \ldots - \theta_q v_{t-q} \end{aligned}$$

Properties of the estimates

Examples:

$$
\begin{aligned}
\text{AR(1):}\quad \mathrm{Var}(\hat{\phi}) &\approx \frac{(1-\phi^2)}{n} \\
\text{AR(2):}\quad \mathrm{Var}(\hat{\phi}_1) &\approx \mathrm{Var}(\hat{\phi}_2) \approx \frac{1-\phi_2^2}{n} \\
\mathrm{Cor}(\hat{\phi}_1, \hat{\phi}_2) &\approx -\frac{\phi_1}{1-\phi_2} = \varrho_1 \\
\text{MA(1):}\quad \mathrm{Var}(\hat{\theta}_1) &\approx \frac{(1-\theta^2)}{n} \\
\text{MA(2):}\quad \mathrm{Var}(\hat{\theta}_1) &\approx \mathrm{Var}(\hat{\theta}_2) \approx \frac{1-\theta_2^2}{n} \\
\mathrm{Cor}(\hat{\theta}_1, \hat{\theta}_2) &\approx -\frac{\theta_1}{1-\theta_2}
\end{aligned}
$$

Properties of the estimates

$$\begin{aligned}
\text{ARMA}(1,1): \quad \text{Var}(\hat{\phi}) &\approx \left(\frac{(1-\phi^2)}{n}\right)\left(\frac{1-\phi\theta}{\phi-\theta}\right)^2 \\
\text{Var}(\hat{\theta}) &\approx \left(\frac{(1-\theta^2)}{n}\right)\left(\frac{1-\phi\theta}{\phi-\theta}\right)^2 \\
\text{Cor}(\hat{\phi},\hat{\theta}) &\approx \frac{\sqrt{(1-\phi^2)(1-\theta^2)}}{1-\phi\theta}
\end{aligned}$$

Remarks:

- For AR(1), variance decreases a $|\phi|$ approaches 1.
- AR(1) is special case of AR(2) with $\phi_2 = 0$, however variance of $\hat{\phi}_1$ is larger.
- Properties of MA model exactly analogous to AR model.
- In ARMA(1, 1) model, the variance can be very large if ϕ and θ are nearly equal.

Properties of the estimates

Remarks:

- CSS is asymptotically equivalent to ML.
- For AR, MM is asymptotically equivalent to ML. However, for MA and ARMA it has considerably larger variance.
- ML estimation is asymptotically efficient under assumption of normal innovations.
- Central limit theorem does not depend on normality assumption, just i.i.d. $(0, \sigma_e^2)$. However, Quasi-ML estimation not efficient.
- ARMA models are more difficult to interpret than pure AR models. Main motivation is to save parameters.
- *Warning:* Parametrizations, estimation, and numerical algorithms differ substantially between packages. See: Newbold, Agiakloglou, Miller (1994). “Adventures with ARIMA Software.” *International Journal of Forecasting*.

Parameter Estimation

Information Criteria

Information criteria

Motivation: Increasing the orders p and q of an ARMA model will always increase the maximized likelihood. However, this is at the “cost” of a less parsimonious model, i.e., with more estimated parameters.

Question: What is a good trade-off between between increasing the likelihood and increasing the model complexity?

Answer: Various methods can be used. For comparing two models, the likelihood ratio test is intuitive. For comparing more than two models, typically information criteria are used.

Idea:

- Penalize the negative log-likelihood by a term that increases with the number of parameters.
- Choose the model with the smallest information criterion.

Information criteria

Definition: For ARMA(p, q) the full parameter vector is $\beta = (\phi_1, \ldots, \phi_p, \theta_1, \ldots, \theta_q)^\top$ with complexity $k = p + q$. Adding an intercept μ, adds one component, i.e, $k = p + q + 1$. Then

$$IC(\beta) \; = \; -2 \cdot \log L(\beta) \; + \; \text{penalty} \cdot k$$

The most common choices for penalties are the Akaike information criterion (AIC), the corrected AIC (AIC_C) of Hurvich and Tsai, the Schwartz Bayesian information criterion (BIC, SBC, or SIC), the Hannan & Quinn criterion (HQ). Penalties:

$$\begin{aligned} \text{AIC}: &\quad 2 \\ \text{AIC}_C: &\quad 2 \cdot \frac{n}{n-k-2} \\ \text{HQ}: &\quad 2 \cdot \log(\log(n)) \\ \text{BIC}: &\quad \log(n) \end{aligned}$$

Information criteria

Remarks:

- Information criteria can be employed both for full-grid and stepwise search, respectively.
- For typical values of n and k, BIC leads to the most parsimonious models, followed by HQ, AIC_C, and AIC.
- If the true process is some finite-order ARMA(p, q) process, then BIC leads to consistent selection of p and q.
- If the true process is not a finite-order ARMA process, an AIC-based selection finds the optimal ARMA model closest to the true model (in the sense of the Kullback-Leibler divergence).
- Maximum likelihood estimation can be affected by numerical problems when choosing too large orders p and q. Hannan and Rissanen algorithm tries to avoid this by some suitable approximation of BIC.

Information criteria

Kullback-Leibler divergence: General measure of distance of distributions. Typically employed for assessing the difference between some true distribution $g(y)$ and a fitted distribution $f(y; \hat{\beta})$.

$$K(g, f_\beta) \; = \; \int \log\left(\frac{g(y)}{f(y;\beta)}\right) \; g(y) \, dy$$

Information criteria

In R:

- `AIC()` is generic function for extracting information criteria from fitted model objects.
- Default `AIC()` method leverages `logLik()` method which extracts log-likelihood and number of estimated parameters k.
- Default penalty is `AIC(obj, k = 2)`, i.e., AIC. Setting `k = log(n)` yields BIC etc.

Furthermore:

- `auto.arima()` from *forecast* implements automatic IC-based model selection for ARIMA models.
- AIC (default), AIC_C, and BIC are available.
- By default `stepwise = TRUE` search is used. If `FALSE` a slower full-grid search is employed.
- By default, d is selected based on KPSS test.

Parameter Estimation

Illustrations of Parameter Estimation

Illustrations of parameter estimation

Convenience function for exploring AR(p) estimation with various estimation methods:

```
R> my_ar <- function(y, order = 1)
+  {
+    rval <- matrix(0, nrow = 3, ncol = order + 1)
+    rownames(rval) <- c("yule-walker", "ols", "mle")
+    colnames(rval) <- c(paste("ar", 1:order, sep = ""), "var")
+
+    for(i in rownames(rval)) {
+      fit <- ar(y, method = i, aic = FALSE, order.max = order)
+      rval[i,] <- c(fit$ar, fit$var.pred)
+    }
+
+    return(rval)
+  }
```

Illustrations of parameter estimation

Apply to simulated AR(1) model with $\phi = 0.9$.

```
R> my_ar(ar1a)
```

```
                ar1    var
yule-walker 0.8867 0.8554
ols         0.8879 0.8394
mle         0.8819 0.8334
```

ML estimation can also be performed via

```
R> arima(ar1a, order = c(1, 0, 0))
```

```
Call:
arima(x = ar1a, order = c(1, 0, 0))

Coefficients:
        ar1  intercept
      0.882     -0.457
s.e.  0.041      0.666

sigma^2 estimated as 0.833:  log likelihood = -160.1,  aic = 326.2
```

Illustrations of parameter estimation

For simulated AR(1) model with $\phi = -0.8$.

```
R> my_ar(ar1b)
```

```
                ar1   var
yule-walker -0.7987 1.328
ols         -0.8001 1.206
mle         -0.8172 1.232
```

```
R> arima(ar1b, order = c(1, 0, 0))
```

```
Call:
arima(x = ar1b, order = c(1, 0, 0))

Coefficients:
         ar1  intercept
      -0.817      0.008
s.e.   0.053      0.056

sigma^2 estimated as 1.23:  log likelihood = -183.3,  aic = 372.7
```

Illustrations of parameter estimation

For simulated AR(2) model with $\phi_1 = 1$ and $\phi_2 = -0.6$.

```
R> my_ar(ar2, order = 2)
```

```
              ar1     ar2    var
yule-walker 1.039 -0.5871 0.7788
ols         1.054 -0.5993 0.7447
mle         1.048 -0.5941 0.7384
```

```
R> arima(ar2, order = c(2, 0, 0))
```

```
Call:
arima(x = ar2, order = c(2, 0, 0))

Coefficients:
        ar1     ar2  intercept
      1.048  -0.594      0.020
s.e.  0.073   0.073      0.144

sigma^2 estimated as 0.738:  log likelihood = -152.8,  aic = 313.6
```

Illustrations of parameter estimation

Attempting auto-selection of p via AIC, works for OLS:

```
R> ar(ar2, method = "ols")
```

```
Call:
ar(x = ar2, method = "ols")

Coefficients:
     1       2
 1.054  -0.599

Intercept: 0.00887 (0.0795)

Order selected 2  sigma^2 estimated as  0.745
```

Illustrations of parameter estimation

Attempting auto-selection of p via AIC, but not quite for ML:

```
R> ar(ar2, method = "mle")
```

```
Call:
ar(x = ar2, method = "mle")

Coefficients:
     1        2        3
 0.960  -0.441  -0.147

Order selected 3  sigma^2 estimated as  0.722
```

Illustrations of parameter estimation

Revisit oil price returns

```
R> data("oil.price", package = "TSA")
R> oil <- diff(log(oil.price))
```

Consider all combinations of $p = 0, 1, 2$, $q = 0, 1, 2$ and zero vs. non-zero mean

```
R> mods <- expand.grid(ar = 0:2, ma = 0:2, mean = c(TRUE, FALSE))
R> mods$aic <- 0
```

Fit all models and extract AIC

```
R> for(i in 1:nrow(mods)) mods$aic[i] <- AIC(
+    arima(oil, order = c(mods$ar[i], 0, mods$ma[i]),
+      include.mean = mods$mean[i]))
```

Illustrations of parameter estimation

```
R> mods
   ar ma  mean    aic
1   0  0  TRUE -501.3
2   1  0  TRUE -511.4
3   2  0  TRUE -514.1
4   0  1  TRUE -514.9
5   1  1  TRUE -514.7
6   2  1  TRUE -512.1
7   0  2  TRUE -514.6
8   1  2  TRUE -515.4
9   2  2  TRUE -513.9
10  0  0 FALSE -502.6
11  1  0 FALSE -513.1
12  2  0 FALSE -515.6
13  0  1 FALSE -516.6
14  1  1 FALSE -516.2
15  2  1 FALSE -514.2
16  0  2 FALSE -516.2
17  1  2 FALSE -515.8
18  2  2 FALSE -514.2
```

Illustrations of parameter estimation

Refit best model

```
R> mods[which.min(mods$aic),]

   ar ma  mean    aic
13  0  1 FALSE -516.6

R> arima(oil, order = c(0, 0, 1), include.mean = FALSE)

Call:
arima(x = oil, order = c(0, 0, 1), include.mean = FALSE)

Coefficients:
        ma1
      0.296
s.e.  0.069

sigma^2 estimated as 0.00669:  log likelihood = 260.3,  aic = -516.6
```

Illustrations of parameter estimation

Do all in one using `auto.arima()`

```
R> library("forecast")
R> auto.arima(oil, ic = "aic", stationary = TRUE,
+     max.p = 2, max.q = 2, max.P = 0, max.Q = 0,
+     stepwise = FALSE, trace = TRUE, approximation = FALSE)
```

```
 ARIMA(0,0,0)            with zero mean     : -502.6
 ARIMA(0,0,0)            with non-zero mean : -501.3
 ARIMA(0,0,1)            with zero mean     : -516.6
 ARIMA(0,0,1)            with non-zero mean : -514.9
 ARIMA(0,0,2)            with zero mean     : -516.2
 ARIMA(0,0,2)            with non-zero mean : -514.6
 ARIMA(1,0,0)            with zero mean     : -513.1
 ARIMA(1,0,0)            with non-zero mean : -511.4
 ARIMA(1,0,1)            with zero mean     : -516.2
 ARIMA(1,0,1)            with non-zero mean : -514.7
 ARIMA(1,0,2)            with zero mean     : -515.8
 ARIMA(1,0,2)            with non-zero mean : -515.4
 ARIMA(2,0,0)            with zero mean     : -515.6
 ARIMA(2,0,0)            with non-zero mean : -514.1
```

Illustrations of parameter estimation

```
 ARIMA(2,0,1)            with zero mean     : -514.2
 ARIMA(2,0,1)            with non-zero mean : -512.1
 ARIMA(2,0,2)            with zero mean     : -514.2
 ARIMA(2,0,2)            with non-zero mean : -513.9



 Best model: ARIMA(0,0,1)            with zero mean

Series: oil
ARIMA(0,0,1) with zero mean

Coefficients:
        ma1
      0.296
s.e.  0.069

sigma^2 estimated as 0.00672:  log likelihood=260.3
AIC=-516.6   AICc=-516.5   BIC=-509.6
```

Parameter Estimation

Bootstrapping ARIMA Models

Bootstrapping ARIMA models

Question: How should inference in ARIMA models be carried out when the asymptotic approximations are not reliable, e.g., for small or heteroskedastic series?

Answer: One alternative approach to asymptotic approximations are bootstrap procedures.

Idea:

- Generate new series $\{Y_t^*\}$ from the "same" distribution as the observed series $\{Y_t\}$ and reestimate the model.
- Repeat series generation and estimation B times.
- This results in an empirical distribution of estimated parameters.
- Standard errors and confidence intervals can be computed from this empirical distribution (instead of an asymptotic distribution).

Bootstrapping ARIMA models

Question: How should $\{Y_t^*\}$ be generated?

Answers:

- Employ estimated ARIMA model but simulate new innovations, either with zero mean and variance $\hat{\sigma}_e^2$ or by resampling $\hat{e}_t$.
- Assume stationarity of Y_t and resample blocks of Y_t.
- ...

In R:

- `arima.boot()` from *TSA* for ARIMA model-based bootstrap.
- `tsbootstrap()` from *tseries* for block bootstrap of stationary series.