Estimating Time-Series Models

Outline

1. Estimating ACVFs and ACFs
2. Estimating parameters of an ARMA and ARIMA model
3. Residual analysis
4. R implementation

Estimating ACVFs and ACFs

Suppose we have N observations on a stationary process, say $x_1, x_2, \ldots, x_N$.

- Then the sample autocovariance at lag k is given by

$$c_k := \widehat{\gamma}(k) = \frac{1}{N}\sum_{t=1}^{N-k}(x_t - \bar{x})(x_{t+k} - \bar{x}),$$

  is the usual estimator for the autocovariance $\gamma(k)$ at lag k. An alternative estimator of $\gamma(k)$ is

$$c'_k = \frac{1}{N-k}\sum_{t=1}^{N-k}(x_t - \bar{x})(x_{t+k} - \bar{x}),$$

- Having estimated the acv.f., we then take

$$r_k := \widehat{\rho}(k) = c_k/c_0$$

  as an estimator for $\rho(k)$.

- Suppose that $x_1, \ldots, x_N$ are observations on independent and identically distributed random variables with arbitrary mean, then we have, (The proof is beyond our discussion)

$$E(r_k) \approx -1/N, \qquad \text{Var}(r_k) \approx 1/N$$

  and that r_k is asymptotically normally distributed under weak conditions.

- We can check for randomness by plotting approximate 95% confidence limits at $-1/N \pm 2/\sqrt{N}$, which are often further approximated to $\pm 2/\sqrt{N}$. Observed values of r_k which fall outside these limits are significantly different from 0 at the 5% level.

Using the correlagram in modeling

The correlogram is helpful in identifying a suitable class of models for a given time series.

- The correlogram of nonstationary seris (e.g., random walks) doesn't come down to zero quickly.
- For stationary series, the correlogram is compared with the theoretical ac.f.s of different ARMA process in order to choose the one which seems to be the 'best' representation.
- The ac.f. of an MA(q) process cuts off at lag q; the ac.f. of an AR(p) process dies out slowly.

Estimating parameters of an ARMA model

- Given the order (p, q) of an ARMA(p, q) process,

  $$X_t - \alpha_1 X_{t-1} - \dots \alpha_p X_{t-p} = Z_t + \beta_1 Z_{t-1} + \cdots + \beta_q Z_{t-q}.$$

  the coefficients $\alpha_1, \dots, \alpha_p, \beta_1, \dots, \beta_q$ are estimated by minimizing the residual sum of squares (RSS)

  $$\min_{\alpha_1,\dots,\alpha_p,\beta_1,\dots,\beta_q} \sum_{t=1}^{n} Z_t^2,$$

- The minimization step for the RSS is usually carried out by numerical optimization, and implemented by commonly used statistical softwares.

Estimating parameters of an ARIMA model

- In Box-Jenkins ARIMA modelling, the general approach is to difference an observed time series until it appears to be come from a stationary process.
- Then the differenced series are estimated by an AR, MA or ARMA model, as described above.

Residual analysis

- When a model has been fitted to a time series, it is advisable to check that the model really does provide an adequate description of the data. As with most statistical models, this is usually done by looking at the residuals, which are generally defined by

$$\text{residual} = \text{observation} - \text{fitted value}.$$

- If we have a good model, then we expect the residual to be random and close to zero. Model validation usually consists of two important steps: plot the residuals a time plot, and calculate the correlogram of the residuals.
- Note that $1/\sqrt{N}$ supplies an upper bound for the standard error of the residual autocorrelations, so that values, which lie outside the range $\pm 2/\sqrt{N}$, are significantly different from zero at the 5% level and give evidence that the wrong form of model has been fitted.

Residual analysis

- Instead of looking at the residual ac.f. one at a time, we may also carry out the portmanteau lack-of-fit test: The test statistic is

$$Q = N \sum_{k=1}^{K} r_{z,k}^2,$$

where N is the sample size of the residual series and K is usualy chosen in the range 15 to 30. If the fitted model is appropriate, then Q should be approximately as χ^2_{K-p-q}, where p and q are the order of AR and MA terms, respectively, in the model.

R implementation

- R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS.
- To download R, go to the website of R-project:

$$http://www.r-project.org$$

click the "CRAN" on the left side, and choose a location close to you. Then you can download and install R in your laptop.
- The following link gives an introduction to R:

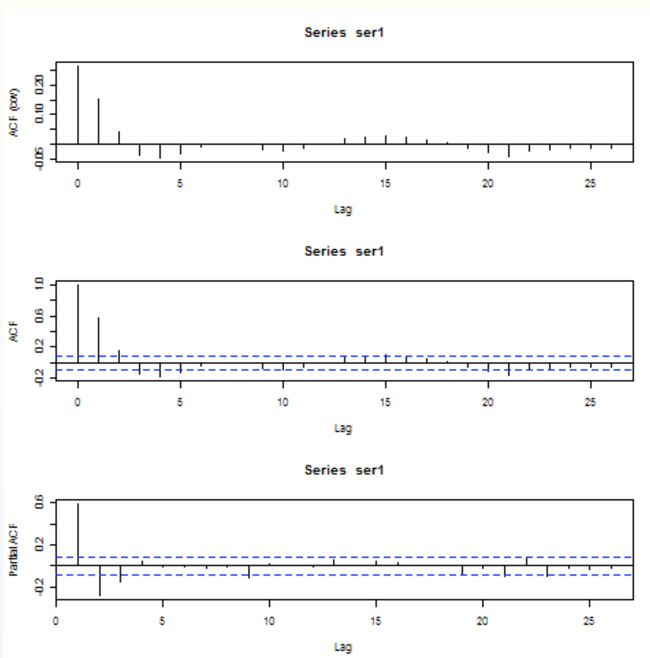
$$http://cran.r-project.org/doc/manuals/R-intro.pdf$$

R function — arima, acf

```
> ? arima.sim ### Simulate from an ARIMA model.
> ser1 <- arima.sim(n = 500, list(ar = c(0.8897, -0.4858),
ma = c(-0.2279, 0.2488)), sd = sqrt(0.1796))
> ? acf ### Compute the sample acvf, acf, or partial acf

> png("fig_Rexample_01.png")
> par(mfrow=c(3,1))
> acf(ser1, type="covariance")
> acf(ser1, type="correlation")
> acf(ser1, type="partial")
> dev.off()
```


Figure 1: The estimated ACVF, ACF and partial ACF



R function — arima

```
> ? arima ### Fit an ARIMA model to a univariate time series.
> fit1<-arima(ser1, order=c(1,0,1))
> fit1
Call: arima(x = ser1, order = c(1, 0, 1))
Coefficients:
         ar1     ma1  intercept
      0.4252  0.2895     0.0067
s.e.  0.0557  0.0520     0.0405
sigma^2 estimated as 0.1639: log likelihood=-257.58, aic=523.16
> names(fit1)
 [1] "coef"  "sigma2"    "var.coef" "mask"    "loglik"   "aic"
 [7] "arma"  "residuals" "call"     "series"  "code"     "n.cond"
[13] "model"
> fit1$coef
        ar1         ma1   intercept
0.425166746 0.289516585 0.006703504
> plot(fit1$resid) ### Plot the residual series
> fit2<-arima(ser1, order=c(2,0,1))
```