STA 137 Homework 5

Camden Possinger

2022-05-27

1. For MA(1) model $x_t = \theta w_{t-1} + w_t$ using the innovation algorithm to show that

$$\theta_{n1} = \frac{\theta\sigma^2}{P_n^{n-1}}, \theta_{nj} = 0, j = 2, 3, ..., n$$

The general strategy to prove these claims is to conduct a proof by induction.

$$\theta_{t,t-j} = \frac{\gamma(t-j) - \sum_{k=0}^{j-1} \theta_{j,j-k}\theta_{t,t-k}P_{k+1}^k}{P_{j+1}^j}$$

When

$$t = n$$
$$n - j = 1$$
$$j = n - 1$$

$$n = 0$$
$$j = -1$$

$$\theta_{01} = \frac{\gamma(1)}{P_n^{n-1}}$$

Since

$$\sum_{k=0}^{-2} \theta_{j,j-k}\theta_{t,t-k}P_{k+1}^k = 0$$

For an MA(1) model:

$$\gamma(1) = \theta\sigma^2$$

$$\theta_{01} = \frac{\theta\sigma^2}{P_n^{n-1}}$$

Next we need to find $\theta_{1,1}$

When

$$t = n$$
$$n - j = 1$$
$$j = n - 1$$

$$n = 1$$
$$j = 0$$

$$\theta_{11} = \frac{\gamma(1)}{P_n^{n-1}}$$

Since

$$\sum_{k=0}^{-1} \theta_{j,j-k}\theta_{t,t-k}P_{k+1}^{k} = 0$$

For an MA(1) model:

$$\gamma(1) = \theta\sigma^2$$

$$\theta_{11} = \frac{\theta\sigma^2}{P_n^{n-1}}$$

By induction $\theta_{n1} = \frac{\theta\sigma^2}{P_n^{n-1}}$

Next we need to prove that $\theta_{nj} = 0, j = 2, 3, ..., n$

First let's prove that $\theta_{n2} = 0$

When

$$t = n$$
$$n - j = 2$$
$$j = n - 2$$

$$n = 0$$
$$j = -2$$

$$\theta_{02} = \frac{\gamma(2)}{P_{-1}^{-2}}$$

Since

$$\sum_{k=0}^{-3} \theta_{j,j-k}\theta_{t,t-k}P_{k+1}^{k} = 0$$

$$P_{-1}^{-2} = \gamma(0)$$

Since

$$\sum_{j=0}^{-3} \theta_{t,t-j}^2 P_{j+1}^j = 0$$

For an MA(1) model:

$$\theta_{02} = \frac{\gamma(2)}{\gamma(0)} = \rho(2) = 0$$

Now let's prove that $\theta_{12} = 0$

When

$$t = n$$
$$n - j = 2$$
$$j = n - 2$$

$$n = 1$$
$$j = -1$$

$$\theta_{12} = \frac{\gamma(2)}{P_0^{-1}}$$

Since

$$\sum_{k=0}^{-2} \theta_{j,j-k}\theta_{t,t-k} P_{k+1}^k = 0$$

$$P_0^{-1} = \gamma(0)$$

Since

$$\sum_{j=0}^{-2} \theta_{t,t-j}^2 P_{j+1}^j = 0$$

For an MA(1) model:

$$\theta_{12} = \frac{\gamma(2)}{\gamma(0)} = \rho(2) = 0$$

By induction $\theta_{n2} = 0$

For the last step we need to prove that $\theta_{n3} = 0$ as well.

When

$$t = n$$
$$n - j = 3$$
$$j = n - 3$$

$$n = 0$$
$$j = -3$$

$$\theta_{03} = \frac{\gamma(3)}{P_{-2}^{-3}}$$

Since

$$\sum_{k=0}^{-4} \theta_{j,j-k}\theta_{t,t-k}P_{k+1}^{k} = 0$$

$$P_{-2}^{-3} = \gamma(0)$$

Since

$$\sum_{j=0}^{-4} \theta_{t,t-j}^{2} P_{j+1}^{j} = 0$$

For an MA(1) model:

$$\theta_{03} = \frac{\gamma(3)}{\gamma(0)} = \rho(3) = 0$$

When

$$t = n$$
$$n - j = 3$$
$$j = n - 3$$

$$n = 1$$
$$j = -2$$

$$\theta_{13} = \frac{\gamma(3)}{P_{-1}^{-2}}$$

Since

$$\sum_{k=0}^{-3} \theta_{j,j-k}\theta_{t,t-k}P_{k+1}^{k} = 0$$

$$P_{-1}^{-2} = \gamma(0)$$

Since

$$\sum_{j=0}^{-3} \theta_{t,t-j}^2 P_{j+1}^j = 0$$

For an MA(1) model:

$$\theta_{13} = \frac{\gamma(3)}{\gamma(0)} = \rho(3) = 0$$

By induction $\theta_{n3} = 0$

Finally also by induction

$$\theta_{nj} = 0, j = 2, 3, ..., n$$

3.10 Let x_t represent the cardiovascualr morality series (cmort) discussed in Example 2.2.

(a) Fit and AR(2) to x_t using linear regression as in Example 3.18

```
model <- cmort %>% ar.ols(order = 2, demean = FALSE, intercept = TRUE)
```

(b) Assuming the fitted model in (a) is the true model, find the forecasts over a four-week horizon x_{n+m}^n for $m = 1, 2, 3, 4$ and the corresponding 95% prediction intervals.

```
forecast(ar(cmort,
            order = 2,
            demean = FALSE,
            intercept = TRUE,
            method = c("ols")),
         level = 95, h = 4) %>% kable
```

	Point Forecast	Lo 95	Hi 95
1979.769	87.59986	76.45777	98.74196
1979.788	86.76349	74.64117	98.88581
1979.808	87.33714	73.35431	101.31997
1979.827	87.21350	72.33079	102.09621

3.21 Generate 10 realizations of length $n = 200$ each of an ARMA(1,1) process with $\phi = .9$,$\theta = .5$ and $\sigma^2 = 1$. Find the MLEs of the three parameters in each case and compare the estimators to the true values.

```
sim_lst <- 1:10 %>% lapply(function(x){

  arima.sim(list("ar" = c(0.9), "ma" = c(0.5), "order" = c(1,0,1)), 200)
})


mle_lst <- sim_lst %>% lapply(function(time_series){
```

```
  model <- time_series %>% arima(order = c(1,0,1), method = "ML")
  mle <- model %>% extract2("coef") %>% extract(c(1,2))
  mle %<>% append(model %>% extract2("sigma2"))
  names(mle)[3] <- "sigma2"
  return(mle)

})

sim_names <- 1:length(mle_lst) %>% sapply(function(index){

  paste0("Simulation ", index)

})

mle_df <- mle_lst %>% as.data.frame %>% t
rownames(mle_df) <- sim_names
mle_df %>% kable
```

	ar1	ma1	sigma2
Simulation 1	0.9408836	0.3600222	1.0285192
Simulation 2	0.9409318	0.4535220	1.0060185
Simulation 3	0.9191400	0.6643022	0.9454327
Simulation 4	0.8642785	0.4446738	0.8582464
Simulation 5	0.8733682	0.5471148	0.9494384
Simulation 6	0.8701835	0.5161107	0.9061250
Simulation 7	0.8816780	0.5986629	1.0111542
Simulation 8	0.8864353	0.5032184	0.9837081
Simulation 9	0.8821936	0.5798898	0.8534479
Simulation 10	0.8659508	0.4864491	0.8929083

Here the estimated values are pretty close to the true values of these parameters. Some over estimate and some under estimate. The true values are:

$$\phi = 0.9$$
$$\theta = 0.5$$
$$\sigma^2 = 1$$

3.22 Generate $n = 50$ observations from a Gaussian AR(1) model with $\phi = .99$ and $\sigma_w = 1$. Using an estimation technique of your choice, compare the approximate asymptotic distribution of your estimate (the one you would use for inference) with the results of a bootstrap experiment (use B = 200).

```
set.seed(42)


dex =  arima.sim(n=50, list(ar=.99,order = c(1,0,0)))

#AR(1) modeling w/ Yule-Walker
fit = ar.yw(dex, order=1)
m = fit$x.mean
phi = fit$ar

round(cbind(fit$x.mean, fit$ar, fit$var.pred), 2) %>% print
```

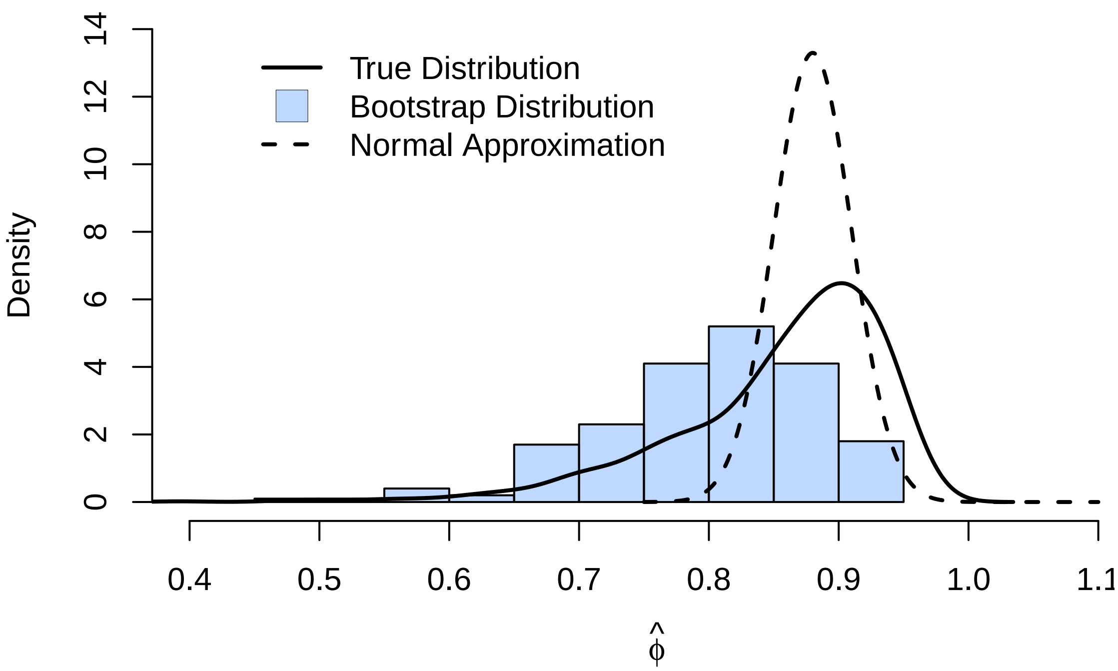
```
##       [,1] [,2] [,3]
## [1,] -0.47 0.88 1.76
```

```
set.seed(101)
phi.yw = rep(NA, 1000)
for (i in 1:1000){
  x =  arima.sim(n=50, list(ar=.99))
  phi.yw[i] = ar.yw(x, order=1)$ar }

nboot = 200
resids = fit$resid[-1]
x.star = dex
phi.star.yw = rep(NA, nboot)

set.seed(102)
for (i in 1:nboot) {
  resid.star = sample(resids, replace=TRUE)
  for (t in 1:49){ x.star[t+1] = m + phi*(x.star[t]-m) + resid.star[t] }
  phi.star.yw[i] = ar.yw(x.star, order=1)$ar
}

culer = rgb(.5,.7,1,.5)
hist(phi.star.yw, 15, main="", prob=TRUE, xlim=c(0.4,1.12), ylim=c(0,14),
     col=culer, xlab=expression(hat(phi)))
lines(density(phi.yw, bw=.02), lwd=2)
u = seq(.75, 1.1, by=.001)
lines(u, dnorm(u, mean=.88, sd=.03), lty=2, lwd=2)
legend(.45, 14, legend=c('True Distribution', 'Bootstrap Distribution','Normal Approximation'),
       bty='n', lty=c(1,0,2), lwd=c(2,0,2),col=1, pch=c(NA,22,NA), pt.bg=c(NA,culer,NA), pt.cex=2.5)
```



Here the bootstrap distribution underestimates the parameter $\hat{\phi}$ compared to the the true distribution and normal approximation but is not far off. With more iterations the bootstrap distribution will become more and more Gaussian.