Time_Series_HW_4

Patrick Kelly

Thursday, May 14, 2015

Assignment Notes:

The daily data is from Illinois Dept of Transporation (IDOT) for I80E 1EXIT (the 2nd data column) - note each data point is an hourly count of the number of vehicles at a specific location on I80E.

Use the daily data for last 2 weeks of June 2013 to develop an ARIMA forecasting model.

Objective is to forecast the hourly counts for July 1.

The actual data file for July 1 is included for you to test your estimate.

Part 1

Use ARIMA(p,d,q) model to forecast. Find the model returned by R auto.arima(). Change the values of p and q and determine the best model using AICc and BIC. Do AICc and BIC select the same model as the best model?

First, I manipulated the data in MS Excel to get it into a usable format.

```r
#import the data
raw_data <- read.csv("~/R/UChicago/Time_Series/I80_EAST_data.csv")
#there were some accidentally included empty rows at the end - clean those up
raw_data <- raw_data[1:384,]
#convert date column to dates
raw_data$Date <- as.character(raw_data$Date)
raw_data$Date <- as.Date(raw_data$Date, "%m/%d/%Y")
#create value of date and time merged together
raw_data$date_time <- as.POSIXct(paste(raw_data$Date,raw_data$Time), format="%Y-%m-%d %H:%M")

#convert data into time series format
data_ts <- ts(raw_data$I80E_1EXIT, frequency = 24)

#subsetting data --- splitting between June and July
june_ts <- window(data_ts, start = c(1,1), end = c(15,24))
july_ts <- window(data_ts, start = c(16,1), end = c(16,24))

#fit the ARIMA(p,d,q) model using auto.arima()
library("forecast")
```

```
## Loading required package: zoo
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
##
## Loading required package: timeDate
## This is forecast 6.0
```

```
(aa_fit <- auto.arima(june_ts, seasonal = FALSE))
```

```
## Series: june_ts
## ARIMA(2,0,3) with non-zero mean
##
## Coefficients:
##          ar1      ar2      ma1      ma2      ma3  intercept
##       1.8088  -0.8853  -0.5348  -0.2671  -0.1157   746.3181
## s.e.  0.0288   0.0287   0.0600   0.0596   0.0654     6.8586
##
## sigma^2 estimated as 13219:  log likelihood=-2220.78
## AIC=4455.56   AICc=4455.88   BIC=4482.77
```

```
#select model based on best AICc and BIC values
#rather than changing p and q manually, I am using the auto.arima() to optimize for these different par
#optimize for AICc
(aa_fit2 <- auto.arima(june_ts, seasonal = FALSE, ic="aicc"))
```

```
## Series: june_ts
## ARIMA(2,0,3) with non-zero mean
##
## Coefficients:
##          ar1      ar2      ma1      ma2      ma3  intercept
##       1.8088  -0.8853  -0.5348  -0.2671  -0.1157   746.3181
## s.e.  0.0288   0.0287   0.0600   0.0596   0.0654     6.8586
##
## sigma^2 estimated as 13219:  log likelihood=-2220.78
## AIC=4455.56   AICc=4455.88   BIC=4482.77
```

```
#optimize for BIC
(aa_fit3 <- auto.arima(june_ts, seasonal = FALSE, ic="bic"))
```

```
## Series: june_ts
## ARIMA(2,0,2) with non-zero mean
##
## Coefficients:
##          ar1      ar2      ma1      ma2  intercept
##       1.8308  -0.9072  -0.5916  -0.3254   746.3649
## s.e.  0.0229   0.0228   0.0488   0.0471     6.9120
##
## sigma^2 estimated as 13327:  log likelihood=-2222.26
## AIC=4456.52   AICc=4456.76   BIC=4479.83
```

Do AICc and BIC select the same model as the best model?

No, as the output above indicates, when the non-seasonal arima model is optimized for AICc versus BIC different models are produces. The fit optimized for AICc produces ARIMA(2,0,3), while the fit optimized for BIC produces ARIMA(2,0,2).

Part 2

Use day of the week seasonal ARIMA(p,d,q)(P,Q,D)s model to forecast for July 1 (which is a Monday)

```r
library(tseries)
#day of week seasonal model
#create new ts object with a frequency that aligns with the day of the week
data_ts_2 <- ts(raw_data$I80E_1EXIT, frequency = (24*7))

#subsetting data --- splitting between June and July
june_ts_2 <- window(data_ts_2, start = c(1,1), end = c(3,24))
july_ts_2 <- window(data_ts_2, start = c(3,25), end = c(3,48))

#fit model
check.aa.fit <- auto.arima(june_ts_2, seasonal = TRUE)
(day.fit <- arima(june_ts, order=c(0,1,2), seasonal = list(order=c(0,1,0), period = (168) )))
```

```
##
## Call:
## arima(x = june_ts, order = c(0, 1, 2), seasonal = list(order = c(0, 1, 0), period = (168)))
##
## Coefficients:
##           ma1      ma2
##       -0.4741  -0.4853
## s.e.   0.0593   0.0586
##
## sigma^2 estimated as 7007:  log likelihood = -1121.66,  aic = 2249.31
```

```r
#forecast
forecast.7.1<-forecast(day.fit,h=24)
#check forecase output
forecast.7.1$mean
```

```
## Time Series:
## Start = c(16, 1)
## End = c(16, 24)
## Frequency = 24
##  [1]  231.35419  140.97855  141.97855  176.97855  352.97855  775.97855
##  [7] 1125.97855 1205.97855 1080.97855  899.97855  909.97855  898.97855
## [13]  926.97855  982.97855 1022.97855 1104.97855 1196.97855 1125.97855
## [19]   17.97855  270.97855  525.97855  534.97855  476.97855  326.97855
```

Part 3

Use hour of the day seasonal ARIMA (p,d,q)(P,D,Q)s model to forecast for the hours 8:00, 9:00, 17:00 and 18:00 on July 1

```r
#hour of the day seasonal model
check.aa.fit <- auto.arima(june_ts, seasonal = TRUE)
(hour.fit <- arima(june_ts, order=c(2,0,1), seasonal = list(order=c(2,0,0), period = (24) )))
```

```
##
## Call:
## arima(x = june_ts, order = c(2, 0, 1), seasonal = list(order = c(2, 0, 0), period = (24)))
##
## Coefficients:
```

```
##          ar1      ar2      ma1    sar1    sar2  intercept
##       1.7922  -0.8685  -0.9146  0.4866  0.1010   743.7286
## s.e.  0.0299   0.0291   0.0257  0.0555  0.0557    13.6793
##
## sigma^2 estimated as 10558:  log likelihood = -2184.12,  aic = 4382.23
```

```
#forecast
forecast2.7.1<-forecast(hour.fit,h=24)
#check forecase output for 8:00, 9:00, 17:00 and 18:00
forecast2.7.1$mean[c(8,9,17,18)]
```

```
## [1] 756.5516 854.0998 933.5026 846.3402
```

Part 4

For the July 1 8:00, 9:00, 17:00 and 18:00 forecasts, which model is better (part 2 or part 3) ?

```
#determine residuals based on the difference between actual and projected

#day of week model
(f1<-as.vector(forecast.7.1$mean[c(8,9,17,18)]))
```

```
## [1] 1205.979 1080.979 1196.979 1125.979
```

```
(a1<-as.vector(july_ts_2[c(8,9,17,18)]))
```

```
## [1] 1233 1110 1142 1129
```

```
abs(f1-a1)
```

```
## [1] 27.021445 29.021445 54.978555  3.021445
```

```
#total error
sum(abs(f1-a1))
```

```
## [1] 114.0429
```

```
#hour of day model
(f2<-as.vector(forecast2.7.1$mean[c(8,9,17,18)]))
```

```
## [1] 756.5516 854.0998 933.5026 846.3402
```

```
(a2<-as.vector(july_ts[c(8,9,17,18)]))
```

```
## [1] 1233 1110 1142 1129
```

```
abs(f2-a2)
```

```
## [1] 476.4484 255.9002 208.4974 282.6598
```

```r
#total error
sum(abs(f2-a2))
```

```
## [1] 1223.506
```

The part 2 (day of week) model is better based on checking the residuals, as displayed above.