Predicting Crude Oil Prices

Bhaumik R Nariya
Industrial And Systems Engineering
Georgia Institute of Technology
Atlanta, GA 30318
bnariya3@gatech.edu

Abstract

Our aim here is to predict the price of Crude Oil. Specifically, we will be looking at the price of Brent Crude Oil which is considered as a benchmark in global oil trade. The dataset that we are going to use is compiled from OPEC, which stands for Organization of Petroleum Exporting Countries and accounts for almost 80% of the global oil reserves. The methods that we will be using here are based on ARIMA model and VAR model. The dataset contains three time series namely Global Oil Demand, Oil Production and nominal price per barrel of Crude Oil. We have yearly data available on these variables from 1980 to 2017. In the end, we show that restricted VAR model, due to low model complexity and its ability to capture inter-dependencies among different time series , outperforms the univariate ARIMA and its unrestricted cousin specially for the short horizon forecasting.

1 Problem Formulation

Significance Predicting the price of crude oil is extremely important as it influences the global economics. For example, countries which rely heavily on petroleum imports can hedge against future price hike if they can predict oil prices accurately. On the other hand, OPEC countries which relay mainly on petroleum exports to sustain their economies can proactively manage their oil production which can help attaining the target export revenues.

Approach We will briefly define the ARIMA and VAR models and the associated parameters. Then, we will divide the dataset into training and testing, estimate model parameters using training set and evaluate forecast performance on testing set. The training set consist of data points from year 1980 to 2012 and the testing set consists of data points from 2013 to 2017.

1.1 Autoregressive Integrated Moving Average (ARIMA)

Definition ARIMA is a generalization of ARMA (autoregressive moving average) model. Here, $\{Y_t\}$ representes the nominal yearly crude oil prices from 1980 to 2017. The ARIMA(p, d, q) model representation is as follows.

$$\left(1 - \sum_{i=1}^{p} \phi_i B^i\right) (1 - B)^d Y_t = \left(1 + \sum_{i=1}^{q} \theta_i B^i\right) \epsilon_t$$

where B is a backward operator, d is the order of difference used to make the time series stationary, p is the autoregressive (AR) order and q is the moving average (MA) order, ϕ_i are AR coefficients and θ_i are MA coefficients. In a more compact way, we can write the above model as follows.

$$\phi(B)(1-B)^d Y_t = \theta(B)\epsilon_t$$

where $\phi(B)$ is an AR polynomial of order p and $\theta(B)$ is an MA polynomial of order q. Also note that ϵ_t is a white noise process with zero mean and variance σ^2 and represents random error which can not be captured in the model. So, we have that $\epsilon_t \sim WN(0, \sigma^2)$.

Stationarity To make the time series data stationary, we first take log transformation as it helps to make the variance constant with respect to time. To select the appropriate difference order, we use the R command `ndiff` from the `ts` library and it suggested to do 1^{st} order differecing. After taking one-lag difference, we plot the transformed time series, we noticed that it seems stationary and the same is also confirmed by the ACF and PACF plots.[1]

Order Selection To select the order (p, d, q) for the ARIMA model, we will use the AIC (Akaike information criterion) which in a broad sense represents the sum of model fitting error and model complexity. Mathematically,

$$AIC(\bar{\phi}, \bar{\theta}, \sigma^2) = -2\ lnL(\bar{\phi}, \bar{\theta}, \sigma^2) + \frac{2(p+q+1)n}{n-p-q-2}$$

The first term here represents data fitting error and the second term represents model complexity.$\bar{\phi}$ represents a vector of all autoregressive coefficients and $\bar{\theta}$ represents a vector of all moving average coefficient and σ^2 is the variance of the error term. n is the total sample size and p and q are AR and MA orders. By iterating through all possible combination of orders, we choose the one which minimizes the AIC the most. The best order is found to be $(1, 0, 1)$. Since we had already taken the difference, the best model has difference order $d = 0$. When $d = 0$, the ARIMA model reduces to ARMA model. We can define ARMA$(1, 1)$ as below.

$$Y_t - \phi_1\ Y_{t-1} = \epsilon_t - \theta_1\ \epsilon_{t-1}$$

Parameter Estimation Once we know the order, we can easily estimate the model parameters with R command `arima` and specifying maximum likelihood approach for parameter estimation. The output of `arima` R command can be summarized in the table below.

Table 1: Output of R command `arima`

Parameter	Estimate (using ML)
AR coefficient, ϕ_1	0.12225
MA coefficient, θ_1	-0.22137
Variance of the Error σ^2	0.06450

Residual Analysis To access the quality of the fitted model, we will look at the residuals, also known as model fitting error i.e. ϵ_t for all time points t. The residual plot shows almost constant variance (except a couple of points) which is in cohort with our model assumption for ARMA. Also, ACF plot shows stationarity and histogram along with normal Q-Q plot shows normality of the error term with some left skewness behavior.

[1] In order to keep the report succinct, we are not showing the ACF and PACF plots but reader can verity the same using the sample code provided in the Appendix

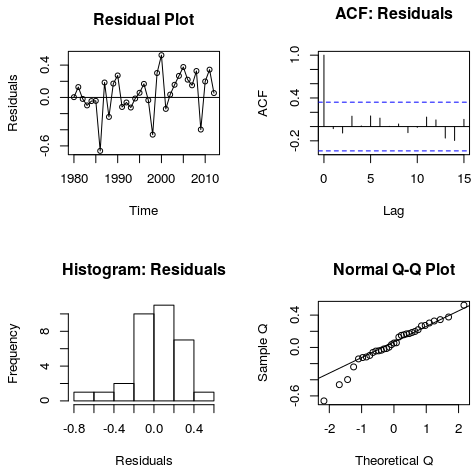

Figure 1: Residual Analysis for ARMA$(1, 1)$

From the residual analysis, we can conclude that our model is captures the temporal dependency reasonably well. Also, the Box-Pierce and Ljung-Box test shows that the residuals are stationary.

Forecasting We do forecasting for the testing period i.e. from 2013 to 2017 and compare it with the original observed values. The blue dotted line in the plot represents 95% confidence interval.

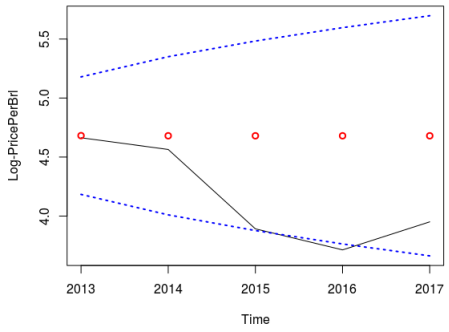

Figure 2: Forecast using ARMA$(1, 1)$

From the figure, we can notice that the model performs well for the immediate future i.e. for the year 2013 and 2014 but as we move futhur into the future the forecast quality reduces. The point-estimate (as red circles in the figure above) overshoots the actual values. Note that in the plot we have log transformed values of Price Per Barrel.

1.2 Vector Autoregressive (VAR)

Definition VAR stands for vector autoregressive model and is a natural extension of univariate autoregressive model. Here, the only difference being that $\{Y_t\}$ is a vector comprising of multiple time series. In our case, it is a 3 dimensional vector of Global Oil Demand measured in thousand barrels per day, Oil Production also in thousand barrels per day and nominal price of crude oil in US dollars/ barrel.

Mathematically, VAR model of order p with deterministic component is represented as below.

$$Y_t = c + \pi_1 Y_{t-1} + \pi_2 Y_{t-2} + ... + \pi_p Y_{t-p} + \Phi D_t + \epsilon_t$$

where c is a vector of constant term with same dimension as Y_t, D_t is a matrix of deterministic components which take care of the trend and seasonality. π_i are square matrix of the size same as the dimension of the multivariate series Y_t. $\epsilon_t \sim WN(0, \Sigma)$ is an error vector of dimension same as Y_t, all the π_is are coefficient-matrix at lag i and they enable to capture interdependencies among the lagged values of different time series. Also, the variance-covariace matrix Σ of ϵ_t helps to capture contemporaneous correlations among different time series.

Order Selection After taking the first order difference of the log-transformed time series (by doing this, we are making the time series stationary as we saw in the univariate case), we will combine them into a vector using R command `ts.union` then we will use order selection criteria to choose the VAR order p. The R command `VARselect` from the `vars` library suggests an order p of 4 as per AIC, Hannan–Quinn information criterion (HQ) , Bayseian Information Criteria (BIC) and Akaike's Final Prediction Error (FPE). So, all the methods suggest lag order p of 4 for VAR model. So, we can represent the data with the following VAR(4) model.

$$Y_t = c + \pi_1 Y_{t-1} + \pi_2 Y_{t-2} + \pi_3 Y_{t-3} + \pi_4 Y_{t-4} + \Phi D_t + \epsilon_t$$

Parameter Estimation First, we will fit unrestricted VAR model which takes into account all the values till lag four for all the three time series i.e. global oil demand, oil production and price per barrel.We can represent the unrestricted model for price per barrel series as below.

$$Y_{t3} = c_3 + (\pi^1_{31} Y_{t1-1} + ... + \pi^4_{31} Y_{t1-4}) + (\pi^1_{32} Y_{t2-1} + ... + \pi^4_{32} Y_{t2-4}) + (\pi^1_{33} Y_{t3-1} + ... + \pi^4_{33} Y_{t3-4}) + \epsilon_{t3}$$

The first set of parenthesis captures the dependence on the lagged values of the first time series through coefficients $\pi^1_{31}, ..., \pi^4_{31}$. The second set of parenthesis does the same with second time series through $\pi^1_{32}, ..., \pi^4_{32}$. The third set of parenthesis captures the dependence on its own lagged values with coefficient $\pi^1_{33}, ..., \pi^4_{33}$. ϵ_{t3} is the third component of the error vector. The above model is unrestricted. On the other hand, if we only consider the statistically significant coefficients and include them in the model, we will get the 'so-called' restricted VAR model. Below is the output of the restricted VAR model[2] for the price time series.

Table 2: Output of Restricted VAR model for Price time series

Coefficient	**Estimate**	**Std. Error**	**t-value**	**Pr(>|t|)**
PricePerBrl.l1	0.43268	0.16795	2.576	$0.01723 **$
GblOilDmd.l1	11.48283	3.67462	3.125	$0.00493 **$
GblOilDmd.l2	−9.46144	4.69694	−2.014	$0.05636*$
GblOilDmd.l3	9.27944	3.09109	3.002	$0.00656 **$
PricePerBrl.l4	0.32438	0.15114	2.146	$0.04314 **$
const	−123.04981	44.68491	−2.754	$0.01159 **$
trend	−0.13270	0.05755	−2.306	$0.03093 **$
Significant codes:	*.05 '**'*	*.1 '*'*		
Residual standard error: *0.1988 on 22 degrees of freedom*				
Multiple R-Squared: *0.9974*			***Adjusted R-squared:*** *0.9966*	
F-statistic: *1224 on 7 and 22 DF*			***p-value:*** $2.2e-16$	

Note that this output is only for the Price time series, we have similar table for both of the remaining time series namely the global oil demand and the oil production. As we can see from the table that all the coefficients are statistically significant either at the 0.05 or at 0.10 significance level. The $p-value$ for the model is very low as shown in table 2 indicating statistical significance of the model

[2]Here, we only show the output from the Restricted VAR. Reader can also interpret the output of the unrestricted VAR in the same manner.

and also the adjusted R-square value is close to 0.99 indicating that 99% of the variability in the data is explained by the model which is much higher than the unrestricted model.

Next, we will forecast for the testing period. The red circles shows the forecasted values using unrestricted VAR while the green circles shows the forecasted values for the restricted VAR model. We can see that restricted VAR model performs better here than the unrestricted one.

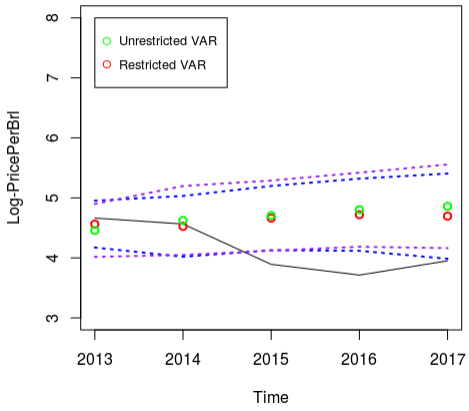

Figure 3: Unrestricted Vs. Restricted VAR

From the figure 3, we can say that both the restricted and unrestricted VAR model overestimates as we move further into the forecasting horizon. But for the next two years i.e. for 2013 and 2014, they predict the price reasonably well.

Of course, in practice this scheme is implemented on a rolling horizon basis and projections are generally limited to one or two years into the future.

2 Conclusions

The following table summarizes the performance of the different time series using two of the most commonly used performance metrics.

Table 3: Forecast Performance of ARIMA, Unrestricted VAR and Restricted VAR

	Two Year Forecast		**Five Year Forecast**	
Model Used	**MSE**	**MAD**	**MSE**	**MAD**
ARMA(1,1)	0.0830	0.0670	0.6482	1.3090
Restricted VAR(4)	0.1517	0.1326	0.7373	1.5379
Unrestricted VAR(4)	0.0763	0.0698	0.6590	1.3303

From the above table, we conclude that for the short term forecast i.e. one-two year into the future, restricted VAR performs the best followed by ARMA(1,1) as per MSE (Mean Squared Error) metric while as per MAD (Mean Absolute Deviation) metric ARMA(1,1) performs best followed by restricted VAR. For the distant future i.e. three-five year into the future, ARMA(1,1) performs the best followed by restricted VAR as per both MSE and MAD metrics. Unrestricted VAR model performs the worst among the three models as it considers the "full" model and has too many parameters to estimate.

Overall, we conclude that ARMA and restricted VAR performs better and they give superior forecast specially when the forecast horizon is short i.e. one or two year into the future.

References

[1] **Brockwell, Peter, and Richard Davis** (2006) *Time Series: Theory and methods.* New York : Springer.

[2] **DOE,US.2018b.** "US energy information administration: independent statistics and analysis.".

[3] **Bosler, Fabian Torben** (2010) "Models for oil price prediction and forecasting." PhD diss. San Diego State University, Department of Mathematics.

[4] **Alquist, Ron, LutzKilian and Robert Vigfusson** (2011) "Forcasting of the Price of Oil." International Finance Discussion Papers, 1022

[5] **Ron, Alquist, KilianLutz, and Vigfusson Robert** (2011) "Forecasting the Prices of Oil." *IDEAS Working Paper Series from RePEc*, 55(1): 869–889.

[6] **OPEC Data and Graphs.** (n.d.) from `https://www.opec.org/opec_web/en/data_graphs/40.html`

[7] **OPEC Global Oil Data.** (n.d.) from `https://asb.opec.org/download/zip2.php`

Appendix R Code

R code can be found on Github along with CSV files.

`https://github.com/bnariya3/Time-Series-Analysis-R-Code-Project`

Also, you can interactively run the code on R Studio Cloud on your browser with pre-installed dependencies.

`https://rstudio.cloud/project/312641`

```
# Read Data Files

GlobalOilDemand = read.csv("GlobalOilDemand.csv", header = T)
OilProd = read.csv("OilProduction.csv", header = T)
PricePerBrl = read.csv("PricePerBarrel.csv", header = T)

# Converting to Multivariate Time Series
GlobalOilDemand.ts = ts(as.numeric(unlist(GlobalOilDemand)), start = 1980,
                        end = 2017, frequency = 1)
OilProd.ts = ts(as.numeric(unlist(OilProd)), start = 1980,
                end = 2017, frequency = 1)
PricePerBrl.ts = ts(as.numeric(unlist(PricePerBrl)), start = 1980,
                    end = 2017, frequency = 1)

data.ts = ts.union(GlobalOilDemand.ts,OilProd.ts, PricePerBrl.ts)
plot(data.ts, type = "l", main = "")
acf(data.ts, mar = c(3.5,3,1.9,0))
pacf(data.ts, mar = c(3.5,3,1.9,0))

# Testing for Stationarity
library(tseries)
adf.test(GlobalOilDemand.ts, alternative = "stationary")
adf.test(OilProd.ts, alternative = "stationary")
adf.test(PricePerBrl.ts, alternative = "stationary")

library(forecast)
#Determine differencing order to achieve stationarity
ndiffs(GlobalOilDemand.ts, alpha = 0.05, test = c("adf"))
ndiffs(OilProd.ts, alpha = 0.05, test = c("adf"))
ndiffs(PricePerBrl.ts, alpha = 0.05, test = c("adf"))

#Log transformation
plot(log(data.ts), type="l",main="")
dGblOilDmd = diff(log(GlobalOilDemand.ts),differences = 1)
dOilProd = diff(log(OilProd.ts),differences = 1)
dPricePBrl = diff(log(PricePerBrl.ts),differences = 1)
ddata.ts = ts.union(dGblOilDmd, dOilProd, dPricePBrl)
```

```
plot(ddata.ts,xlab="time",main="",type="l")
acf(ddata.ts, mar=c(3.5,3,1.9,0))
pacf(ddata.ts, mar=c(3.5,3,1.9,0))

## Data preparation for models: Testing Vs Training
data=data.ts
n = nrow(data)

## Training data: 1980 to 2012
data.train=data[1:(n-5),]
## Test data: 2013 to 2017
data.test=data[(n-4):n,]
ts_GblOilDmd = ts(log(data.train[,"GlobalOilDemand.ts"]),start=1980, freq=1)
ts_OilProd = ts(log(data.train[,"OilProd.ts"]),start=1980, freq=1)
ts_PricePerBrl = ts(log(data.train[,"PricePerBrl.ts"]),start=1980, freq=1)
ts_GblOilDmd2 = ts(log(data.test[,"GlobalOilDemand.ts"]),start=2013, freq=1)
ts_OilProd2 = ts(log(data.test[,"OilProd.ts"]),start=2013, freq=1)
ts_PricePerBrl2 = ts(log(data.test[,"PricePerBrl.ts"]),start=2013, freq=1)

## Univariate ARIMA model
final.aic = Inf
final.order = c(0,0,0,0)
for (p in 1:6) for (d in 0:1) for (q in 1:6) for(s in 0:1){
  current.aic = AIC(arima(ts_PricePerBrl, order=c(p, d, q), seasonal = list(order=c(0,s,0),per
  if (current.aic < final.aic) {
    final.aic = current.aic
    final.order = c(p, d, q,s)

  }
}

# > final.order
# > [1] 1 0 1 1

model.arima = arima(ts_PricePerBrl, order=c(1,0,1),seasonal = list(order=c(0,1,0), period=1),

## Residual analysis
par(mfrow=c(2,2))
plot(resid(model.arima), ylab='Residuals',type='o',main="Residual Plot")
abline(h=0)
acf(resid(model.arima),main="ACF: Residuals")
hist(resid(model.arima),xlab='Residuals',main='Histogram: Residuals')
qqnorm(resid(model.arima),ylab="Sample Q",xlab="Theoretical Q")
qqline(resid(model.arima))

Box.test(model.arima$resid, lag = (1+4+1), type = "Box-Pierce", fitdf = (1+4))
Box.test(model.arima$resid, lag = (1+4+1), type = "Ljung-Box", fitdf = (1+4))

#Predictions versus actual
plot(forecast(model.arima,h=5))
fore = forecast(model.arima,h=5)
fore=as.data.frame(fore)
write.csv(fore,file = "ARIMAForecast.csv")
point.fore = ts(fore[,1],start=2013, freq=1)
lo.fore = ts(fore[,4],start=2013, freq=1)
up.fore = ts(fore[,5],start=2013, freq=1)

ymin=min(c(log(unlist(PricePerBrl)[(n-4):n]),lo.fore))
ymax=max(c(log(unlist(PricePerBrl))[(n-4):n],up.fore))

plot(ts(log(as.numeric(unlist(PricePerBrl)[(n-4):n])),start=2013, freq=1), ylim=c(ymin,ymax),
points(point.fore,lwd=2,col="red")
lines(lo.fore,lty=3,lwd= 2, col="blue")
```

```
lines(up.fore,lty=3,lwd= 2, col="blue")

##
data.train = cbind(ts_PricePerBrl, ts_GblOilDmd, ts_OilProd)
data.test = cbind(ts_PricePerBrl2, ts_GblOilDmd2, ts_OilProd2)
##
library(vars)
##VAR Model with Deterministic Components
##Model Selection
VARselect(data.train, lag.max = 20,type="both")$selection

## Model Fitting: Unrestricted VAR
model.var=VAR(data.train, p=4,type="both")
summary(model.var)

## Model Fitting: Restricted VAR
model.var.restrict=restrict(model.var)
summary(model.var.restrict)

pred = predict(model.var.restrict, n.ahead=5, ci=0.95)[[1]]$ts_PricePerBrl
point.pred = ts(pred[,1],start=2013, freq=1)
ForecastR.VAR = as.data.frame(point.pred)
write.csv(ForecastR.VAR,file = "Forecast_Res_Var.csv")

lo.pred = ts(pred[,2],start=2013, freq=1)
up.pred = ts(pred[,3],start=2013, freq=1)
pred.f = predict(model.var,n.ahead=5, ci=0.95)[[1]]$ts_PricePerBrl
point.pred.f = ts(pred.f[,1],start=2013, freq=1)

ForecastUnrst.VAR = as.data.frame(point.pred.f)
write.csv(ForecastUnrst.VAR,file = "Forecast_Unres_Var.csv")


lo.pred.f = ts(pred.f[,2],start=2013, freq=1)
up.pred.f = ts(pred.f[,3],start=2013, freq=1)

ymin=min(c(log(unlist(PricePerBrl)[(n-4):n]),lo.pred,lo.pred.f))
ymax=max(c(log(unlist(PricePerBrl))[(n-4):n],up.pred,up.pred.f))


plot(ts(log(as.numeric(unlist(PricePerBrl)[(n-4):n])),start=2013, freq=1), ylim=c(3,8),
     ylab="Log-PricePerBrl",type="l",main="")
points(point.pred,lwd=2,col="red")
lines(lo.pred,lty=3,lwd= 2, col="blue")
lines(up.pred,lty=3,lwd= 2, col="blue")
legend(x=2013,y = 8,legend=c("Unrestricted VAR","Restricted VAR"),pch = c(1,1)
       ,pt.cex = 1, cex = 0.8, col = c("green","red"))

points(point.pred.f,lwd=2,col="green")
lines(lo.pred.f,lty=3,lwd= 2, col="purple")
lines(up.pred.f,lty=3,lwd= 2, col="purple")
```