

# A MS Example

Tianyang Xie, Jie Ding

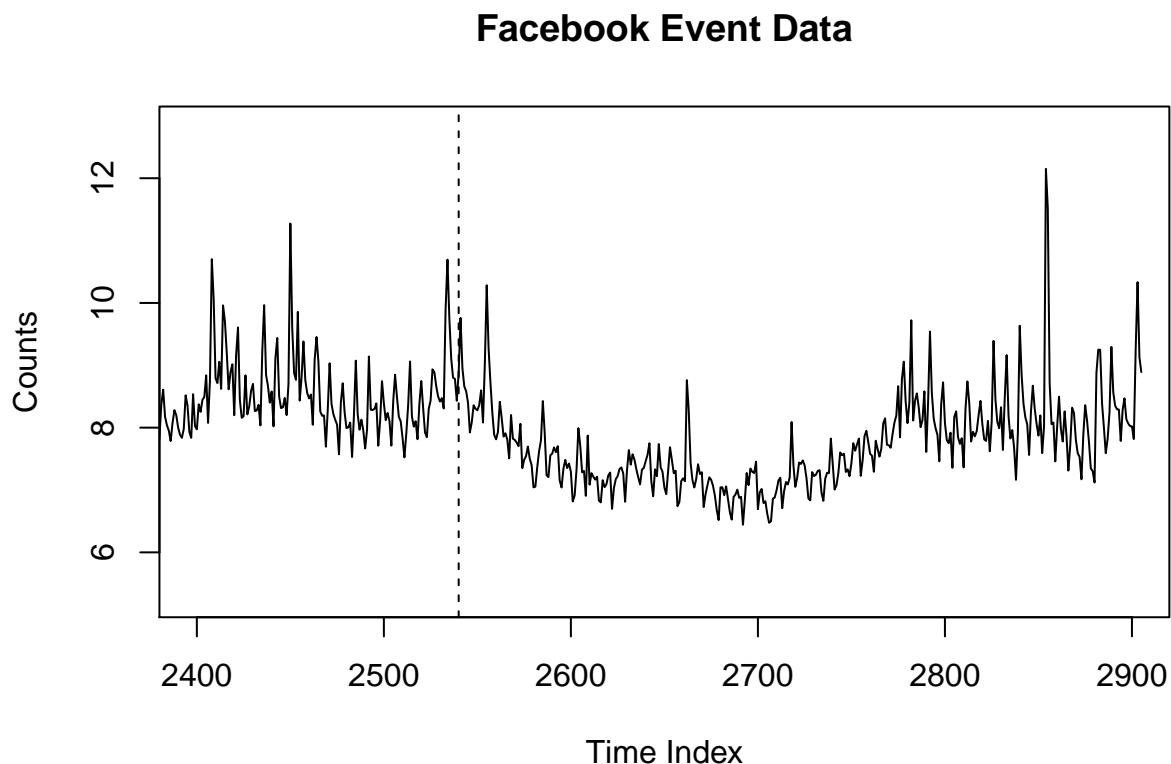
09/04/2020

In this short vignette, we will present a real data example of MS' usage. We first load the 'tsms' package.

```
library(tsms)
#> Loading required package: gridExtra
#> Loading required package: parallel
```

The data set 'prop' comes from the example in the original paper of Facebook 'Prophet' model. It shows the daily counts for the number of events created on Facebook during the dates from 12/10/2007 to 01/20/2016. Weekly and yearly seasonality are strongly suspected, as part of the data shows below. And say, our primary our task is to forecast the 300 steps after the time index 2540.

```
load('C:/Users/Tianyang/Documents/Long-short term/Rawcode/package_principle/data/prophet.RData')
plot(prop$y,type='l',xlim=c(2400,2900),main='Facebook Event Data',ylab='Counts',xlab='Time Index')
abline(v=2540,lty=2)
```



To be able to use the MS modeling procedure, we first need to extract the trend or the intercept from the

series, because the current version of MS can only deal with ‘detrended’ data.

```
X = prop$y
intercept = mean(X[1:2540])
train = (X - intercept)[1:2540]
```

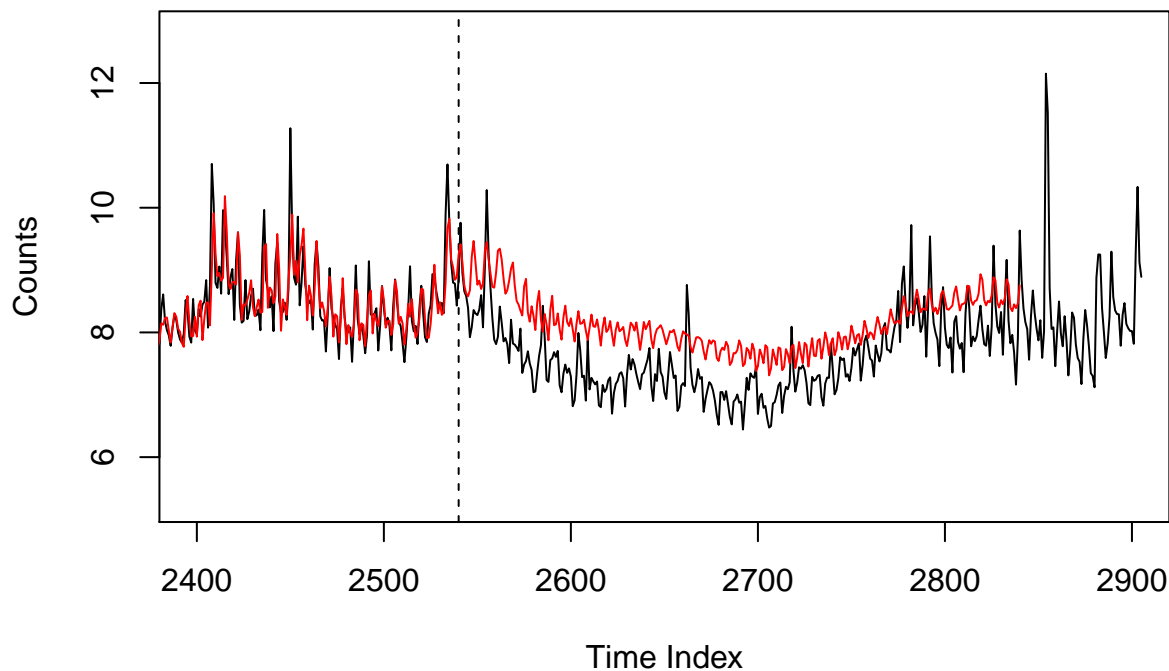
To accommodate the multiple seasonality effects and perform a forecasting, we can employ the MS modeling procedure by setting up the number of potential seasonality terms to be 3, by ‘r=3’. We also allow the maximum lag order of regular AR and MA components to be up to 5, by ‘p\_range=c(0:5);q\_range=c(0:5)’, so that the procedure can select from a enough set of parameters’ combination.

```
U=600
p_range = c(0:5)
q_range = c(0:5)
r=3
pred_t = 300

result = MS(train,U=U,p_range=p_range,q_range=q_range,r=r,pred_t=pred_t)

plot(X,type='l',xlim=c(2400,2900),main='Facebook Event Data',ylab='Counts',xlab='Time Index')
abline(v=2540,lty=2)
lines(result$prediction + intercept,col='red')
```

## Facebook Event Data



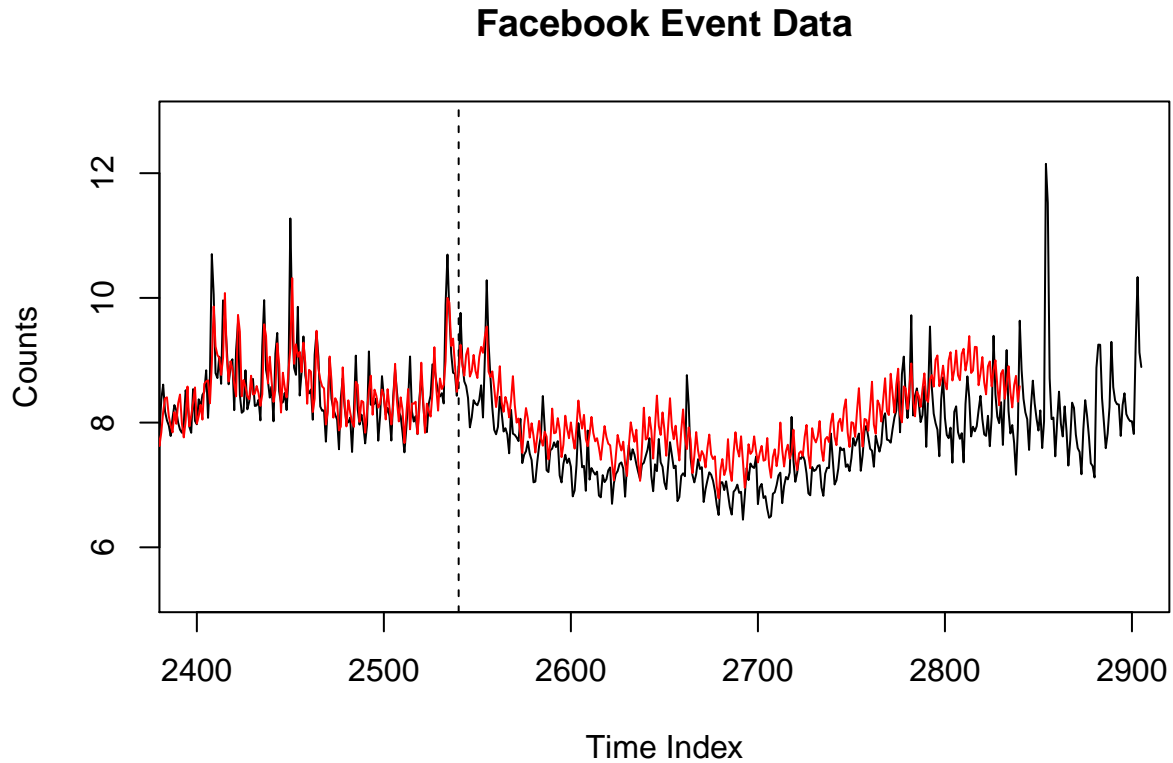
We can further improve the flexibility of the procedure by further increasing the bandwidth of each seasonality term. By setting ‘blur.out=c(6,6)’, we allow the each seasonality period includes the AR and MA components with lag order range from lag-6 to lag+6. Larger value in both elements in ‘blur.out’ provides more fitting power, but also absorbs more risk of overfitting.

```

result = MS(train,U=U,p_range=p_range,q_range=q_range,r=r,pred_t=pred_t,blur.out=c(6,6))

plot(X,type='l',xlim=c(2400,2900),main='Facebook Event Data',ylab='Counts',xlab='Time Index')
abline(v=2540,lty=2)
lines(result$prediction + intercept,col='red')

```



In the end, to present the performance of the MS procedure, we evaluate it on a rolling basis, in which the MS will be evaluated 7 times for forecasting the following 200 steps.

```

# parameters specification
U=600
p_range = c(0:5)
q_range = c(0:5)
r=3
pred_t = 200
rpt = 7

# fitting & prediction
prediction.vec = rep(0,pred_t*rpt)
X.vec = tail(prop$y,pred_t*rpt)

for (i in 1:rpt){
  X = as.matrix(prop$y[1:(nrow(prop)-(rpt+1-i)*pred_t)])
  result = MS(scale(X,scale=F),U=U,p_range=p_range,q_range=q_range,r=r,
                blur.out=c(6,6),pred_t=pred_t)
  prediction = result$prediction + mean(X)
}

```

```

prediction.vec[(1+(i-1)*pred_t):(i*pred_t)] = tail(prediction,pred_t)
}

# plotting
par(mar = c(4,3,1,1) + 0.1,mgp=c(3,0.5,0))
plot(X.vec,type='l',col='blue',ylab='',xlab='',xaxt='n',yaxt='n',bty='n')
box(lwd=2)
axis(1,at=seq(from=0,to=1400,by=200),tck=0.01)
axis(2,at=seq(from=6,to=13,by=1),tck=0.01)
abline(v=seq(from=0,by=pred_t,length.out=rpt),lty=2)
lines(prediction.vec,col='red')
legend('topright',legend=c('Data','MS'), lwd = 2
      ,col=c('blue','red'),lty=rep(1,2),cex=1)
mtext(text = "Time Index",side = 1,line = 1.8)
mtext(text = 'Value',side = 2, line = 1.8)

```

