ensemble_IC2

Saier Gong (sg3772)

8/8/2020

IMPORT DATA

```
data<-read.csv(file="US-Coronavirus-data.csv",header = TRUE) %>%
#data<-data %>%
  #mutate(date=as.character(date)) %>%
  mutate(date=as.Date(date)) %>%
  #mutate(date=paste(month(date),"/",day(date))) %>%
  #mutate(date=as.Date(date,"%m-%d")) %>%
  #mutate(S=N-C) %>%
  mutate(daily.D=D-c(0,D[-length(D)])) %>%
  filter(!is.na(C)) %>%
  filter(date>=as.Date('2020-02-29'))


D<-data %>%
  select(date,I,D,daily.D) %>%
  filter(date<=as.Date("2020-07-31"))

trans<-data$I[-1]/data$I[-nrow(data)] #from March to May
#daily<-data$daily.D
```

MODEL

In this model, we use the same method as in the Imperial College model 1.

However, in model1, we just assume the length of time window is one week, 7 days. In this model, we will use APE to choose the best length of time window. Others are the same as model1.

```
#########likelihood


ppd<-function(k){
  #ppd<-c()
  n<-c()
  p<-c()
  ii<-c()
  ln_a<-c()

  for (i in 1:(length(D$daily.D)-1)) {
    if (i<=k) {
```

```
      next

    }else{

      daily<-D$daily.D[1:i]
      shape=1+sum(tail(daily,k))

      rate<-1/5
      for (s in (i-k+1):i) {
        rate<-rate+mean(daily[1:s-1])

    }
      scale<-1/rate

      n<-append(n,shape)
      p<-append(p,mean(daily) *scale / (1+mean(daily) *scale))

      #ppd<-append(ppd,dnbinom(x=D$daily.D[i+1],size=n,prob=p))
      #ppd[which(ppd==0)]<-10^-320
      ii<-append(ii,i)

      #ln_zuhe<-(D$daily.D[i+1]+shape-1)*(log(D$daily.D[i+1]+shape-1)-1) - (D$daily.D[i+1]*(log(D$daily

      stirl<-function(n){
        ##stirling appproximation to get ln(n!)
        ## return the approximation of  ln(n!)
        return(n*log(n)-n)
      }

      ln_zuhe<-stirl(D$daily.D[i+1]+shape-1)-stirl(D$daily.D[i+1])-stirl(shape-1)
      ln_a<-append(ln_a,ln_zuhe)

    }

  }

  return(list(prob=p,i=ii,ln_zuheshu=ln_a,n=n))

}


ape<-function(k){
  sum<-0
  #sum((-1) * log(ppd(k)$prob))
  haha<-ppd(k)
  for (i in haha$i) {
    #sum<-sum+haha$ln_zuheshu[i-k]+D$daily.D[i+1]*log(haha$prob[i-k])+haha$n[i-k]*log(1-haha$prob[i-k])
    sum<-sum+(-1)*(dnbinom(D$daily.D[i+1],size=haha$n[i-k],prob=haha$prob[i-k],log=TRUE))
  }

  return(sum)

}
```

```r
####optimize APE
## k between 2 and t/2
k.best<-which.min(map_dbl(2:floor(length(D$daily.D)/2),function(k) ape(k)))+1
k.best
```

```
## [1] 2
```

```r
#k.best=2
```

We can get the best length of time window is 2 days.

Then, we will use the most recent 2 days in training set to estimate the R_t.

```r
###estimate Rt
##first set tao=2
##6.30+7.1
###prior of Rt.tao = gamma(1,1/5)
#set.seed(1)
#shape=1
#rate=1/5
Rt<- tail(trans,1)
mu<-mean(tail(trans,2))
v<-var(tail(trans,2))
rate=mu/v
shape=mu*rate

#daily<-append(daily,rep(0,7))
daily<-D$daily.D
l<-length(daily)

###posterior of Rt,tao
R<-Rt
A<-c()
#####MCMC
for (i in 1:2000) {
  #1st given Rt
  for (j in 0:(k.best-1)) {
    daily[l-j]<-rpois(1,lambda = Rt * mean(head(daily,l-j-1)) )
  }

  #daily[l-0]<-rpois(1,lambda = Rt * mean(head(daily,l-1)) )

  shape<-shape+sum(tail(daily,k.best))

  for (j in 0:(k.best-1)) {
    rate<-rate+mean(head(daily,l-j-1))#+mean(head(daily,l-0-1))
  }


  Rt<-rgamma(1,shape=shape,rate=rate)

  R<-append(R,Rt)
  A<-rbind(A,tail(daily,k.best))
```

```
}



R.post<-mean(tail(R,500))
R.post
```

```
## [1] 1.013499
```

```
w.exam.post.last<-colMeans(A)
```

FORECASTING

In this model, we will first use the estimated $R_t.post$ to do deaths forecasting.

```
####forcast of the following 7 days of daily deaths

t<-7
w<-c()

#set.seed(0)
for (iter in 1:2000) {
  daily<-D$daily.D
  l<-length(daily)

  for (i in (l+1):(l+t)) {
    d<-rpois(1,R.post* (mean(head(daily,i-1))) )
    daily[i]<-d
  }

#daily

  total.new<-map_dbl(1:length(daily),function(x) sum(head(daily,x)))

  w.fore<-tail(total.new,t)
  w<-rbind(w,w.fore)

}

answer=colMeans(tail(w,500))

temp<-seq(from=as.Date('2020-08-01'),by='day',length.out = t)

pred=tibble(date=temp,prediction=answer)

#(pred[1:7,]);(pred[8:14,]);(pred[15:21,]);(pred[22:28,])
pred
```

```
## # A tibble: 7 x 2
##   date       prediction
```

```
##   <date>          <dbl>
## 1 2020-08-01    154826.
## 2 2020-08-02    155839.
## 3 2020-08-03    156851.
## 4 2020-08-04    157863.
## 5 2020-08-05    158876.
## 6 2020-08-06    159886.
## 7 2020-08-07    160896.
```

```
#a<-data %>% filter(month(date)==6) %>%select(D)
#mean((a$D-answer)^2)
```

MSE & RMSE

Since we have the best length of time window is 2 days, so we can just estimate the daily death of the recent 2 days. And then we will remove these 2 days from our training set, and use the new training set with the same method to get the best length and estimate other days in the last week. We just repeat the steps till we finish estimating all the deaths in the previous week.

```
daily<-D$daily.D
l<-length(daily)
daily[(l-(k.best-1)):l]<-w.exam.post.last
total<-map_dbl(1:length(daily),function(x) sum(head(daily,x)))
tail(total,k.best)
```

```
## [1] 152369.8 153379.4
```

```
mse.last<-mean((tail(total,k.best)-tail(D$D,k.best))^2)
mse.last
```

```
## [1] 115086
```

```
D<-head(D,(nrow(D)-2))
trans<-head(trans,(length(trans)-2))

k.best.middle<-which.min(map_dbl(2:floor(length(D$daily.D)/2),function(k) ape(k)))+1
k.best.middle
```

```
## [1] 2
```

```
Rt<-tail(trans,1)
mu<-mean(tail(trans,2))
v<-var(tail(trans,2))
rate=mu/v
shape=mu*rate

#daily<-append(daily,rep(0,7))
daily<-D$daily.D
l<-length(daily)
```

```r
###posterior of Rt,tao
R<-Rt
A<-c()
#####MCMC
for (i in 1:2000) {
  #1st given Rt
  for (j in 0:(k.best.middle-1)) {
    daily[l-j]<-rpois(1,lambda = Rt * mean(head(daily,l-j-1)) )
  }

  #daily[l-0]<-rpois(1,lambda = Rt * mean(head(daily,l-1)) )

  shape<-shape+sum(tail(daily,k.best.middle))

  for (j in 0:(k.best.middle-1)) {
    rate<-rate+mean(head(daily,l-j-1))#+mean(head(daily,l-0-1))
  }


  Rt<-rgamma(1,shape=shape,rate=rate)

  R<-append(R,Rt)
  A<-rbind(A,tail(daily,k.best.middle))


}



R.post<-mean(tail(R,500))
w.exam.post.middle<-colMeans(A)


daily[(l-(k.best.middle-1)):l]<-w.exam.post.middle


library(purrr)
total<-map_dbl(1:length(daily),function(x) sum(head(daily,x)))
tail(total,k.best.middle)
```

```
## [1] 149555.3 150544.6
```

```r
mse.middle<-mean((tail(total,k.best.middle)-tail(D$D,k.best.middle))^2)
mse.middle
```

```
## [1] 401526.1
```

```r
D<-head(D,(nrow(D)-2))
trans<-head(trans,length(trans)-2)

k.best.first<-which.min(map_dbl(2:floor(length(D$daily.D)/2),function(k) ape(k)))+1
k.best.first
```

```
## [1] 2
```

```r
Rt<-tail(trans,1)
mu<-mean(tail(trans,2))
v<-var(tail(trans,2))
rate=mu/v
shape=mu*rate

#daily<-append(daily,rep(0,7))
daily<-D$daily.D
l<-length(daily)

###posterior of Rt,tao
R<-Rt
A<-c()
#####MCMC
for (i in 1:2000) {
  #1st given Rt
  for (j in 0:(k.best.first-1)) {
    daily[l-j]<-rpois(1,lambda = Rt * mean(head(daily,l-j-1)) )
  }

  #daily[l-0]<-rpois(1,lambda = Rt * mean(head(daily,l-1)) )

  shape<-shape+sum(tail(daily,k.best.first))

  for (j in 0:(k.best.first-1)) {
    rate<-rate+mean(head(daily,l-j-1))#+mean(head(daily,l-0-1))
  }


  Rt<-rgamma(1,shape=shape,rate=rate)

  R<-append(R,Rt)
  A<-rbind(A,tail(daily,k.best.first))


}



R.post<-mean(tail(R,500))
w.exam.post.first<-colMeans(A)


daily[(l-(k.best.first-1)):l]<-w.exam.post.first


library(purrr)
total<-map_dbl(1:length(daily),function(x) sum(head(daily,x)))
tail(total,k.best.first)
```

```
## [1] 147969.8 148975.3
```

```r
mse.first<-mean((tail(total,k.best.first)-tail(D$D,k.best.first))^2)
mse.first
```

```
## [1] 222669.1
```

```r
D<-head(D,(nrow(D)-2))
trans<-head(trans,length(trans)-2)

k.best.1<-which.min(map_dbl(2:floor(length(D$daily.D)/2),function(k) ape(k)))+1
k.best.1
```

```
## [1] 2
```

```r
Rt<-tail(trans,1)
mu<-mean(tail(trans,2))
v<-var(tail(trans,2))
rate=mu/v
shape=mu*rate

#daily<-append(daily,rep(0,7))
daily<-D$daily.D
l<-length(daily)

###posterior of Rt,tao
R<-Rt
A<-c()
#####MCMC
for (i in 1:2000) {
  #1st given Rt
  for (j in 0:(k.best.1-1)) {
    daily[l-j]<-rpois(1,lambda = Rt * mean(head(daily,l-j-1)) )
  }

  #daily[l-0]<-rpois(1,lambda = Rt * mean(head(daily,l-1)) )

  shape<-shape+sum(tail(daily,k.best.1))

  for (j in 0:(k.best.1-1)) {
    rate<-rate+mean(head(daily,l-j-1))#+mean(head(daily,l-0-1))
  }


  Rt<-rgamma(1,shape=shape,rate=rate)

  R<-append(R,Rt)
  A<-rbind(A,tail(daily,k.best.1))


}
```

```r
R.post<-mean(tail(R,500))
w.exam.post.1<-colMeans(A)


daily[(l-(k.best.middle-1)):l]<-w.exam.post.1


library(purrr)
total<-map_dbl(1:length(daily),function(x) sum(head(daily,x)))
tail(total,k.best.1)
```

```
## [1] 145958.4 146963.7
```

```r
mse.1<-mean((tail(total,1)-tail(D$D,1))^2)
mse.1
```

```
## [1] 1.647372
```

```r
mse<-mse.last+mse.middle+mse.first+mse.1
mse
```

```
## [1] 739282.9
```

```r
(rmse<-sqrt(mse))
```

```
## [1] 859.8156
```