final_UCLA

Saier Gong + Zhijie Huang

8/8/2020

IMPORT DATA

In this model, we set $N = 60757563$ the number of people that have tested from the website (https://www.cdc.gov/coronavirus/2019-ncov/cases-updates/testing-in-us.html).

The training dataset contains the data from March to the end of July.

```
N<-60757563

data<-read.csv(file="US-Coronavirus-data.csv") %>%
  #mutate(S=N-C) %>%
  filter(!is.na(D)) %>%
  filter(month(date)<=7)

#data<-tail(data.full,7)
data<-data %>%
  mutate(date=as.Date(date)) %>%
  filter(month(date)>=3)
data$X<-1:nrow(data)
```

MODEL

```
sir_equations <- function(time, variables, parameters) {

  with(as.list(c(variables, parameters)), {

    dS <- (-1) * beta * (I+E) * S/N
    dE <-  beta * (I+E) * S/N -delta * E
    dI <-  mu * delta * E - (1-alpha) * gamma * I - alpha * rho * I
    dR <-  (1-alpha) * gamma * I
    dD <-  alpha * rho * I

    return(list(c(dS, dE, dI, dR, dD)))
  })
}
```

```
sir_1 <- function( beta,delta,mu,gamma,alpha,rho,times,S0,E0,I0,R0,D0) {
  require(deSolve) # for the "ode" function
```

```
# the differential equations:
  sir_equations <- function(time, variables, parameters) {
    with(as.list(c(variables, parameters)), {

    dS <- (-1) * beta * (I+E) * S/N
    dE <-  beta * (I+E) * S/N -delta * E
    dI <-  mu * delta * E - (1-alpha) * gamma * I - alpha * rho * I
    dR <-  (1-alpha) * gamma * I
    dD <-  alpha * rho * I

    return(list(c(dS, dE, dI, dR, dD)))
  })
  }

  parameters_values <- c(beta=beta,delta=delta,mu=mu,gamma=gamma,alpha=alpha,rho=rho)

# the initial values of variables:
  initial_values <- c(S = S0, E=E0,I = I0, R = R0, D=D0)

# solving
  out <- ode(initial_values, times, sir_equations, parameters_values)

# returning the output:
  as.data.frame(out)

}
```

NOTE: in this model, since we trained the data beginning in March, we have to give the E state a initial value instead of simply setting it zero. Therefore, we suppose E0 follows $U(0, 0.02N)$, so we set $E0 = \frac{0.02N}{2}$.

```
E0=N*0.02/2
#E0=0
error<-function(beta,delta,mu,gamma,alpha,rho){
  S0=328200000-1
  E0=E0
  I0=data$I[1]
  R0=data$R[1]
  D0=data$D[1]


  times<-data$X

  add.D<-data$D-c(0,data$D[-length(data$D)])

  predictions<-sir_1(beta=beta,delta=delta,mu=mu,gamma=gamma,alpha=alpha,rho=rho,
      times = times,
      S0=S0,E0=E0,I0=I0,R0=R0,D0=D0)

  add.D.pre<-predictions$D-c(0,predictions$D[-length(predictions$D)])

  #ssr<-sqrt(mean(sum((predictions$I-data$I)^2+
  #         (predictions$R-data$R)^2+(predictions$D-data$D)^2)))
  #ssr<-sum((predictions$D-data$D)^2+(predictions$R-data$R))
```

```
  ssr<-sum((predictions$D-data$D)^2)#+sum((add.D-add.D.pre)^2)
  #ssr<-sum((predictions$R-data$R)^2)
  #ssr<-mean((data$I-predictions$I)^2) + sum((data$D-predictions$D)^2)
  return(ssr)
}

#error(beta=beta,delta=delta,mu=mu,gamma=gamma,alpha=alpha,rho=rho)
```

Optimize

```
fn<-function(par){
  error(beta=par[1],delta=par[2],mu=par[3],gamma=par[4],alpha=par[5],rho=par[6])
}

best<-optim(par = c(0.7,1/4,0.3,0.55,0.06,0.1),fn=fn)
```

```
## Loading required package: deSolve
```

```
#lower = c(0,1/14,0,1/14,0,1/21),upper = c(2,1/2,1,1,0.08,1),method = "L-BFGS-B"

beta.best<-best$par[1]
delta.best<-best$par[2]
mu.best<-best$par[3]
gamma.best<-best$par[4]
rho.best=best$par[6]
alpha.best=best$par[5]


prediction.best<-sir_1(beta=beta.best,delta=delta.best,mu=mu.best,gamma=gamma.best,alpha=alpha.best,rho=
      S0=328200000-1,E0=E0,I0=data$I[1],R0=data$R[1],D0=data$D[1],
      times = data$X)
#prediction.best$time<-as.Date(prediction.best$time,origin=data$date[1])
```

MSE & RMSE

```
w.exam<-tail(prediction.best$D,7)
w.exam
```

```
## [1] 147254.5 148075.2 148893.4 149709.1 150522.4 151333.4 152141.9
```

```
mse<-mean((w.exam-tail(data$D,7))^2)
rmse<-sqrt(mse)
mse
```

```
## [1] 810429.5
```

```
rmse
```

```
## [1] 900.2386
```

FORECASTING

```
w.fore<-tail(sir_1(beta=beta.best,delta=delta.best,mu=mu.best,gamma=gamma.best,alpha=alpha.best,rho=rho
      S0=328200000-1,E0=E0,I0=data$I[1],R0=data$R[1],D0=data$D[1],
      times = 1:(length(data$X)+7) )$D,7)

w.fore
```

```
## [1] 152948.0 153751.7 154553.0 155351.9 156148.5 156942.7 157734.6
```