МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
им. М. В. Ломоносова
МЕХАНИКО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ
КАФЕДРА ТЕОРИИ ВЕРОЯТНОСТЕЙ

КУРСОВАЯ РАБОТА

«Применение SEIRD модели для моделирования распространения COVID-19 в России»

Студента 3-го курса
Булавко Егора Павловича

Научный руководитель:
д.ф.-м.н., профессор
Яровая Елена Борисовна

Москва 2021

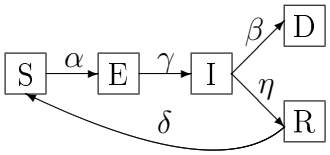
Содержание

Описание модели

Используется SEIRD модель, вся популяция делится на 5 групп:

- S - Susceptibles - люди, подверженные заражению
- E - Exposed - люди, находящиеся в инкубационном периоде, т.е. у которых еще не проявились симптомы и которые еще не заразны
- I - Infected - инфицированные люди, которые могут заражать других
- R - Recovered - выздоровевшие люди
- D - Dead - умершие люди

При этом численность популяции $N = S(t) + E(t) + I(t) + R(t) + D(t)$ остается постоянной, т.е. в модели не учитывается естественная смертность. Поведение модели можно представить в виде схемы:



Дифференциальные уравнения, описывающие поведение модели:

$$\begin{cases} \frac{dS}{dt} = -\frac{\alpha}{N} S \cdot I + \delta R \\ \frac{dE}{dt} = \frac{\alpha}{N} S \cdot I - \gamma E \\ \frac{dI}{dt} = \gamma E - \beta I - \eta I \\ \frac{dR}{dt} = \beta I - \delta R \\ \frac{dD}{dt} = \eta I \end{cases} \tag{1}$$

Начальные условия:

$$S(0) = 0.3 * population; I(0) = 1; E(0) = R(0) = D(0) = 0$$

Где $population = 146171015$ - население России.
Предполагаем, что количество заражений прямо пропорционально произведению количества инфицированных на количество людей, подверженных заражению.

Описание параметров модели

- N - численность населения
- α - параметр заражения, для нормировки в уравнениях делится на N
- γ - параметр, отвечающий за то, насколько долго люди находятся в инкубационном периоде
- β- параметр выздоровления
- η - параметр смертности
- δ - параметр, отвечающий за то, какое количество переболевших людей могут заболеть вновь

Так же в модели учитываются изменения, вызванные введением ограничительных мер: в моменты времени t_1 и t_2, которые можно интерпретировать как начало и конец карантина соответственно, параметр заражения α начинает возрастать линейно с параметром ζ_1 в момент времени t_1 и линейно убывать в момент времени t_2 с параметром ζ_2:

$$\alpha(t) = \begin{cases} \alpha_1, & 0 \leq \text{t} \leq t_1 \\ \max(\alpha_2, \alpha_1 + \zeta_1 \cdot (t - t_1)), & t_1 \leq \text{t} \leq t_2 \\ \min(\alpha_3, \alpha_2 - \zeta_2 \cdot (t - t_2)), & t_2 \leq \text{t} \end{cases}$$

Описание других работ

В статье [1] авторы применяли похожую модель для моделирования второй волны во Франции и Италии. В их работе использовалась SEIR модель, в которой люди не могут заболеть повторно.

$$\begin{cases} \frac{dS}{dt} = -\frac{\alpha}{N} S \cdot I \\ \frac{dE}{dt} = \frac{\alpha}{N} S \cdot I - \gamma E \\ \frac{dI}{dt} = \gamma E - \beta I \\ \frac{dR}{dt} = \beta I \end{cases} \tag{2}$$

С аналогичными начальными условиями для популяций Италии и Франции.

Моделирование производилось так же с помощью разностных уравнений с интервалом $t = 1$ день, были использованы данные университета Джона Хопкинса. Для моделирования того, что параметры могут изменяться во времени, авторы использовали стохастический подход: на каждом шаге параметры менялись

$$\kappa(t) = |\kappa_0 + \sigma\xi(t)|$$

$$\kappa(t) \in \{\alpha(t), \gamma(t), \beta(t)\}$$

$$\xi(t) \sim N(0, 1)$$

$$\alpha_0 = 1; \beta_0 = 0.37; \gamma_0 = 0.27$$

Для моделирования возникновения второй волны параметр α_0 меняется во времени два раза: в момент начала карантина он становится равным 0.25, а в момент снятия ограничений снова 1 (В моей работе этот параметр изменялся не скачками, а линейно). Моделирование производилось несколько раз, после чего результаты усреднялись.

В работе [2] для моделирования эпидемии в Европе использовалась модель epidemic Renormalisation Group (eRG), которая представлят из себя применение подобных SIR модели для разных групп население (популяция не считается однородной).

$$\frac{d\alpha_i}{dt} = \gamma_i \alpha_i \left(1 - \frac{\alpha_i}{a_i}\right) + \sum_{j \neq i} \frac{k_{ij}}{n_{mi}} (e^{\alpha_i - \alpha_j} - 1)$$

где $\alpha_i(t) = ln(\mathscr{A}_i(t))$, $\mathscr{A}_i$ - количество инфицированных в регионе i на миллион, k_{ij} - количество людей, переходящих из региона i в регион j за неделю в миллионах (моделирование производилось с шагом в неделю), γ_i параметр заражения для i-го региона, n_{mi} - население в i-ом регионе в миллионах, a_i - натуральный логарифм итогового количества заражений в конце эпидемии.
Для моделирования второй волны изменяются параметры a_i и γ_i, полученные для первой волны.

Описание моделирования

Для моделирования использовался язык R. Моделирование модели производилось с шагом $t = 1$ день. В этом случае дифференциальные уравнения можно переписать в форме разностных:

$$\begin{cases} S(t+1) - S(t) = -\frac{\alpha(t)}{N} S(t)I(t) + \delta R(t) \\ E(t+1) - E(t) = \frac{\alpha(t)}{N} S(t)I(t) - \gamma E(t) \\ I(t+1) - I(t) = \gamma E(t) - \beta I(t) - \eta I(t) \\ R(t+1) - R(t) = \beta I(t) - \delta R(t) \\ D(t+1) - D(t) = \eta I(t) \end{cases} \tag{3}$$

В качестве оценки использовалась среднеквадратичная ошибка:

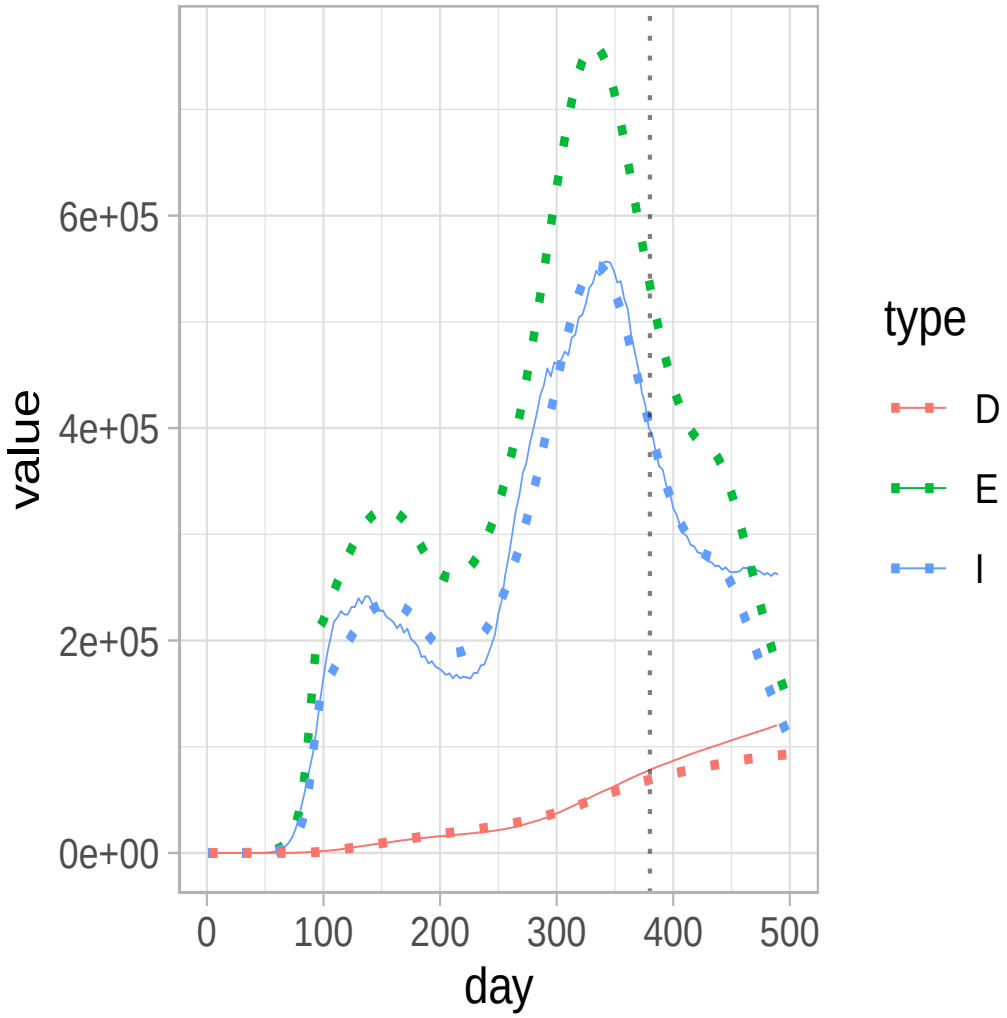
$$Err = \frac{1}{N^2} \sum_{t=1}^{n} (I(t) - \hat{I}(t))^2$$

Где I, D, R - реальные данные, а $\hat{I}$, $\hat{D}$, $\hat{R}$ - данные модели. Коэффициент $\frac{1}{N^2}$ добавлен для нормировки.
Данные были взяты из `https://github.com/CSSEGISandData/COVID-19`
В модели было взято $N = S(0) = 0.3 \cdot population$, где $population = 146748590$ - население России. Поиск оптимальных параметров производился сначала вручную, для поиска первого приближения, и далее с помощью метода Нелдера - Мида, реализованного в R.

Результаты моделирования

В результаты были получены параметры:

α_1	α_2	α_3	β	γ	η	ζ_1	ζ_2	t_1	t_2	δ
0.767	0.415	0.947	0.37	0.27	0.00077	0.0367	0.0013	89	189	0.0047



Пунктиром на графике обозначены данные модели, сплошной линией - реальные данные.

Для подбора параметров модели использовались данные до момента времени, отмеченного вертикальной пунктирной линией на графике.

Код на языке R

```
library(dplyr)
library(tidyr)
library(stringr)
library(ggplot2)
library(ggthemes)
library(deSolve)
library(optimr)

population <- 146748590 * 0.3 #0.3
lock_start <- 89 #23 march 90
lock_end <- 167 #9 june 170



get_data_russia <- function() {
  data_confirmed <- read.csv("https://github.com/CSSEGISandData/COVID-19/
    ↪ raw/master/csse_covid_19_data/csse_covid_19_time_series/time_
    ↪ series_covid19_confirmed_global.csv")
  data_recovered <- read.csv("https://github.com/CSSEGISandData/COVID-19/
    ↪ raw/master/csse_covid_19_data/csse_covid_19_time_series/time_
    ↪ series_covid19_recovered_global.csv")
  data_deaths <- read.csv("https://github.com/CSSEGISandData/COVID-19/raw
    ↪ /master/csse_covid_19_data/csse_covid_19_time_series/time_series_
    ↪ covid19_deaths_global.csv")
  data_confirmed <- pivot_longer(data_confirmed, cols = seq(5, ncol(data_
    ↪ confirmed), 1), names_to = "Date")
  data_recovered <- pivot_longer(data_recovered, cols = seq(5, ncol(data_
    ↪ recovered), 1), names_to = "Date")
  data_deaths <- pivot_longer(data_deaths, cols = seq(5, ncol(data_deaths
    ↪ ), 1), names_to = "Date")
  data_confirmed <- data_confirmed %>% filter(Country.Region == "Russia")
  data_recovered <- data_recovered %>% filter(Country.Region == "Russia")
  data_deaths <- data_deaths %>% filter(Country.Region == "Russia")
  data_confirmed <- data_confirmed[10:nrow(data_confirmed), ]
  data_recovered <- data_recovered[10:nrow(data_recovered), ]
  data_deaths <- data_deaths[10: nrow(data_deaths), ]
  data_russia <- tibble(I = data_confirmed$value - data_recovered$value -
    ↪  data_deaths$value,
                        R = data_recovered$value,
                        S = population - data_confirmed$value,
                        D = data_deaths$value,
                        cumsum = data_confirmed$value,
                        total = data_confirmed$value,
                        day = seq(1, nrow(data_confirmed)))
```

```
  gather(data_russia, type, value, I:total, factor_key=TRUE)
}

plot_data <- function(data, t) {
  ggplot(data = data %>% filter(type == t), aes(x = day, y = value, col =
     ↪  type)) +
  geom_point(size = 0.1)
}

plot_model <- function(data, model, t) {
  ggplot(data = data %>% filter(type == t), aes(x = day, y = value, col =
     ↪  type)) +
    geom_line(size = 0.2) +
    geom_line(data = model %>% filter(type == t), aes(x = day, y = value,
       ↪ col = type),
              linetype = 'dotted', size = 1) + xlab("day") +
          ylab("value") + geom_vline(xintercept = 380, linetype = 'dotted'
             ↪ , size = 0.5, alpha = 0.5) +
    ggtheme)

}

theme_set(theme_light())

plot_cs <- function(data, model) {
  ggplot(data = data, aes(x = day, y = value)) + geom_line(size = 0.2) +
    geom_line(data = model, aes(x = day, y = value), linetype = 'dotted',
              size = 1)
}

data_russia <- get_data_russia()
plot_data(data_russia)

lfn2 <- function(p) {
  out <- SEIRD(S0, E0, I0, R0, D0, alpha1 = p[1] ,
               alpha2 = p[2] ,
               alpha3 = p[3] ,
               beta1 = p[4],
               gamma1 = p[5],
               eta1 = p[6] ,
               n_days = 390, #388
               zeta1 = p[7] ,
               zeta2 = p[8] ,
               lock_start = p[9],
               lock_end = p[10],
               delta1 = p[11] )
```

```
  rss <-
        (sum(((data_russia %>% filter(type == 'I'))$value -
          (out %>% filter(type == 'I'))$value) ^ 2)) / (population *
             ↪ population)
  print(rss)
  return(rss)
}


SEIRD <- function(S0, E0, I0, R0, D0, alpha1, alpha2, alpha3, beta1,
    ↪ gamma1,
                  eta1, n_days, zeta1, zeta2, lock_start, lock_end, delta1)
                     ↪  {
  out <- tibble(day = 1, S1 = S0 + E0, S = S0, E = E0, I = I0, R = R0, D
      ↪ = D0,
                cumsum = I0)

  total <- I0
  alpha = alpha1
  for (i in 2:n_days) {
    S0i <- S0
    E0i <- E0
    I0i <- I0
    R0i <- R0
    D0i <- D0

    if (i < lock_start) alpha = alpha1
    if (i >= lock_start && i < lock_end) alpha = max(alpha2, alpha - zeta1
       ↪ )
    if (i >= lock_end) {
      alpha = min(alpha3, alpha + zeta2)

    }
    if (i >= 430) delta1 = 0

    S0 <- max(0, S0i - alpha * S0i * I0i * 1 / population + delta1 * R0i)
    E0 <- max(0, E0i + alpha * I0i * S0i * 1 / population - gamma1 * E0i)
    I0 <- max(0, I0i + gamma1 * E0i - beta1 * I0i - eta1 * I0i)
    R0 <- max(0, R0i + beta1 * I0i - delta1 * R0i)
    D0 <- max(0, D0i + eta1 * I0i)
    total <- I0 + D0 + R0
    out <- out %>% add_row(day = i, S1 = S0, S = S0 + E0, E = E0, I = I0,
       ↪ R = R0,
                           D = D0, cumsum = total)
  }
  return(gather(out, type, value, S1:cumsum, factor_key=TRUE))
}
```

```
S0 <- population
E0 <- 0
I0 <- 2
R0 <- 0
D0 <- 0

out <- SEIRD(S0, E0, I0, R0, D0, alpha1 = 0.7673478 ,
             alpha2 = 0.4152731 ,
             alpha3 = 0.9476324 ,
             beta1 = 0.370029,
             gamma1 = 0.27,
             eta1 = 0.0007751027 ,
             n_days = 500,
             zeta1 = 0.03674178 ,
             zeta2 = 0.001360923 ,
             lock_start = 89,
             lock_end = 189,
             delta1 = 0.004786315 )

p= c(0.7673478,
     0.4152731,
     0.9476324,
     0.370029,
     0.27,
     0.0007751027,
     0.03674178,
     0.001360923,
     89,
     189,
     0.004786315)

upper = c(2, 2, 2, 1, 1, 1, 1, 1, 100, 200, 0.01)
lower = c(0, 0, 0, 0, 0, 0, 0, 0, 50, 150, 0)

S_ = out %>% filter(type == 'S') + out %>% filter(type == 'E')
plot_model(data = data_russia, model = out, c('I', 'E', 'D'))

meth0 <- c("Nelder-Mead")
res1 <- opm(p,
            lfn2, "grfwd", method=meth0)
res1
```

Так же код доступен на Github

Литература

1. Modelling the second wave of COVID-19 infections in France and Italy via a Stochastic SEIR model, Davide Faranda and Tommaso Alberti
   https://arxiv.org/pdf/2006.05081.pdf

2. Second wave COVID-19 pandemics in Europe: a temporal playbook, Giacomo Cacciapaglia, Corentin Cot, Francesco Sannino
   https://www.nature.com/articles/s41598-020-72611-5

3. Epidemic Modelling: An Introduction, D.J. Daley and J.Gani, Cambridge University press.