Report

Yuanzhi Yu, Zongchao Liu, Sitong Cui, Mengyu Zhang

1. Introduction

2. Method

2.1 Data Clean

2.2 Logistic Curve Model

The function for logisitc curves can be defined as

$$f(t) = \frac{K}{1 + \exp\{-b(t-c)\}}, \tag{2.1}$$

where t is the days since the first infection; K is the upper bound; b is growth rate, and c is the mid-point.

With log-transformation, we get

$$\log\left(\frac{K-N}{N}\right) = a - rt, \tag{2.2}$$

where N_t is the number of cases at t, $a = bc$, $r = b$.

For developing logistic curve model for each country data we apply the following two steps:

Step 1: estimate the upper bound

Step 2: fit logistic curve using gradient descent optimization algorithm

2.2.1 Upper Bound Estimation

Generally, K is estimated by solving a system of equations

$$\begin{aligned} \log\frac{K-N_1}{N_1} &= a - rt_1 \\ \log\frac{K-N_2}{N_2} &= a - rt_2 \\ \log\frac{K-N_3}{N_3} &= a - rt_3 \end{aligned} \tag{2.3}$$

(t_1, N_1), (t_2, N_2) and (t_3, N_3) are the starting point, midpoint and final point in a time sires data respectively. Consequently, we get the formula for K

$$\hat{K} = \frac{2N_1N_2N_3 - N_2^2\left(N_1 + N_3\right)}{N_1N_3 - N_2^2},\ 2t_2 = t_1 + t_3 \tag{2.4}$$

However, $\hat{K}$ could be negtive when N_2 is same as or close to N_1 , so we also use the properties of gradient at midpoint to estimate K.

When $\Delta t \to 0$, $\frac{dN}{dt} \approx \frac{\Delta N}{\Delta t}$, so we get a estimated midpoint $(t_m, \hat{N}_m)$ by

$$\max\left(\frac{N_2 - N_1}{t_2 - t_1}, \frac{N_3 - N_2}{t_3 - t_2}, ..., \frac{N_n - N_{n-1}}{t_n - t_{n-1}}\right). \tag{2.5}$$

Therefore, estimated K would be

$$\hat{K} = 2\hat{N}_m$$

2.2.2 Optimization

Based on this linear relationship between t and cases in formula (2.2), the loss function that we want to minimize can be defined as

$$\begin{aligned} L(t; \hat{a}, \hat{r}, \hat{K}) &= \frac{1}{2}||y - \hat{a} + \hat{r}t||^2 \\ &= \frac{1}{2}|| \log\left(\frac{\hat{K} - N_t}{N_t}\right) - \hat{a} + \hat{r}t||^2, \end{aligned} \tag{2.6}$$

where $\hat{K}$ is estimated upper bound in formula (2.4). The gradient is

$$\nabla L(t; \hat{a}, \hat{r}) = \begin{pmatrix} -\sum_{i=1}^{n}(y_i - \hat{a} + \hat{r}t_i) \\ \sum_{i=1}^{n} t_i(y_i - \hat{a} + \hat{r}t_i) \end{pmatrix} \tag{2.7}$$

Parameters update based on

$$(\hat{a}_i, \hat{r}_i)^T = (\hat{a}_{i-1}, \hat{r}_{i-1})^T - \alpha \nabla L(t; \hat{a}_{i-1}, \hat{r}_{i-1}) \tag{2.8}$$

where α is step length.

The gradient descent optimization algorithm's steps are

Step 1: set starting value $(a, r)^{(0)}$, tolerance ϵ, $i = 1$ and $\alpha = 1$.

Step 2: $L(t; a_i, r_i) - L(t; a_{i-1}, r_{i-1}) \leq \epsilon$, quit and solution is $(a, r)^{(i)}$. Otherwise, to step 3.

Step 3: new estimates is updated by formula (2.8), $\alpha = \alpha/2$ and i = i+1. Turn to step 2.

2.3 EM . . .

3. Results

4. Conclusions