Incorporation of treatment compartment into the fundamental SIR model: an exploration of the treatment effectiveness and its impact on the model dynamics

Titaporn Janjumratsang 23-2102-254

Introduction

Epidemics can be modelled by divide population into distinct compartments and can be expressed using ordinary differential equations (IDEs), this results in deterministic compartmental models. The simplest compartmental model, Kermack-McKendrick SIR model (Carvalho and Sebastián Gonçalves, 2021), consists of three compartments: susceptible, infected, and recovered. For this exercise, the model was extended by implementing "Treated" (T) compartment representing a portion of infected individuals undergoing treatment allowing them to recover at a faster rate than untreated infected individuals, these individuals could also be conditioned to be less infectious in the modified model.

For our model, demographic turnover and migration were not considered, and all infections are assumed to end with recovery. Therefore, population size was defined to be constant (closed system) with no birth or death. Reinfection of recovered individuals was also not considered, i.e. therefore, life-long immunity after recovery was assumed for all individuals. Due to the assumptions made, the model works best for small population for mild transmittable diseases over short period of time e.g. for influenza, mumps, and measles in developed world where deaths from these diseases are rare due to possible treatments available and the individuals develop immunity after recovering from the disease.

Please see R-code generated for the exercise at the end of the report.

Method

The model consists of following system of 4 governing equations:

$$\frac{dS}{dt} = -\beta \cdot S \cdot I - \mu \cdot S \cdot T$$

$$\frac{dI}{dt} = (\beta \cdot I + \mu \cdot T) \cdot S - (\gamma + \theta) \cdot I + \epsilon \cdot T$$

$$\frac{dT}{dt} = \theta \cdot I - (\epsilon + \delta) \cdot T$$

$$\frac{dR}{dt} = \gamma \cdot I + \delta \cdot T$$

Symbol	Description	Value
β	infection rate for untreated individuals	1.5
δ*	recovery rate for treated infected individuals	2
ϵ	rate at which treated individuals revert to the infected compartment	0
γ	recovery rate for untreated infected individuals	0.1
μ*	infection rate for treated individuals	0.1
θ	treatment initiation rate for infected individuals	0.4

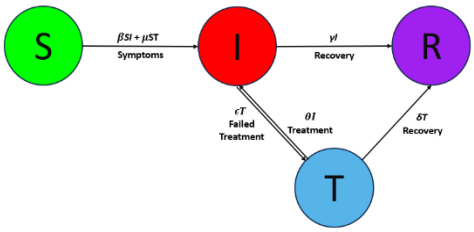

Figure 1: Governing equations of the proposed SITR model (left), the parameters (middle), flow diagram of proposed SITR model (right). *parameters varied in this report*

This model accounts for infected individuals getting treatment at the rate θ, the treatment resulting in a different recovery rate (δ) than those who didn't receive treatment (γ), a different rate of infection for treated individuals (μ) was also accounted for to model possible reduction in the infectiousness of the treated individuals. Furthermore, the rate of possible reversal of treated individuals back to infected compartment (ϵ) was incorporated into the model to account for the treatment failing and for individuals who did not complete the course of treatment (flow diagram figure 1). Although for this report, ϵ was set to 0. The initial conditions were set such that S= 0.999999, I= 0.000001, R=0 and T=0. The parameters values are set based on existing papers (Rafiq et al., 2022), the internet, and some are estimated.

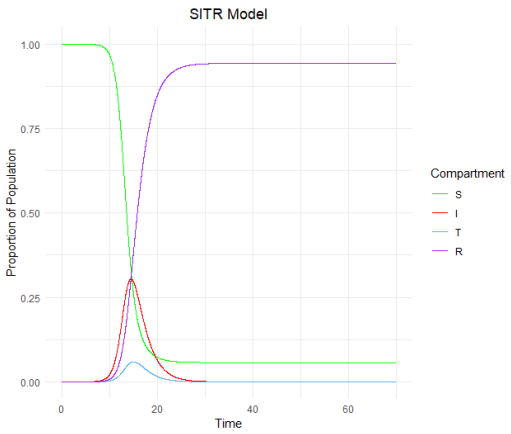

Figure 2: Resulting SITR model with added T compartment as proposed in figure 1.

A model simulation output with the added T compartment is illustrated in figure 2.

Observations & Model analysis

It was observed from the model that effectiveness of the treatment is also highly dependent on the original disease dynamics. Depending on the disease natural rate of recovery and infection, the change in treated individuals' recovery and infection rates can be effective in preventing the spread of the disease to the rest of the population or have little effect. However, the general trend found for the parameters set is as follows.

Treated individuals → faster rate of recovery + same rate of infection:

By varying δ (the rate of recovery for treated infected individuals) while keeping other parameters constant, as δ gets larger, equilibrium is reached faster and R at equilibrium is lower. The peaks of I and T become smaller (this is not illustrated in the report due to constraints in number of pages but can be reproduced in part 2 of the provided code) For our parameter values, the equilibrium is reached in such way that the disease dies out before the entire population becomes infected and gain immunity, this effect is stronger as δ gets larger.

Treated individuals → same rate of recovery + lower rate of infection:

Lowering μ (rate of infection for treated individuals) can also be used to model quarantining of infected individuals. To investigate μ, other parameters were kept constant with different μ values, the lower the μ, the lower and less sharp the I peak (can also be reproduced in part 2 of the provided code) and the lower the R is at equilibrium (figure 3 right). The equilibrium is also reached slower as μ decreases.

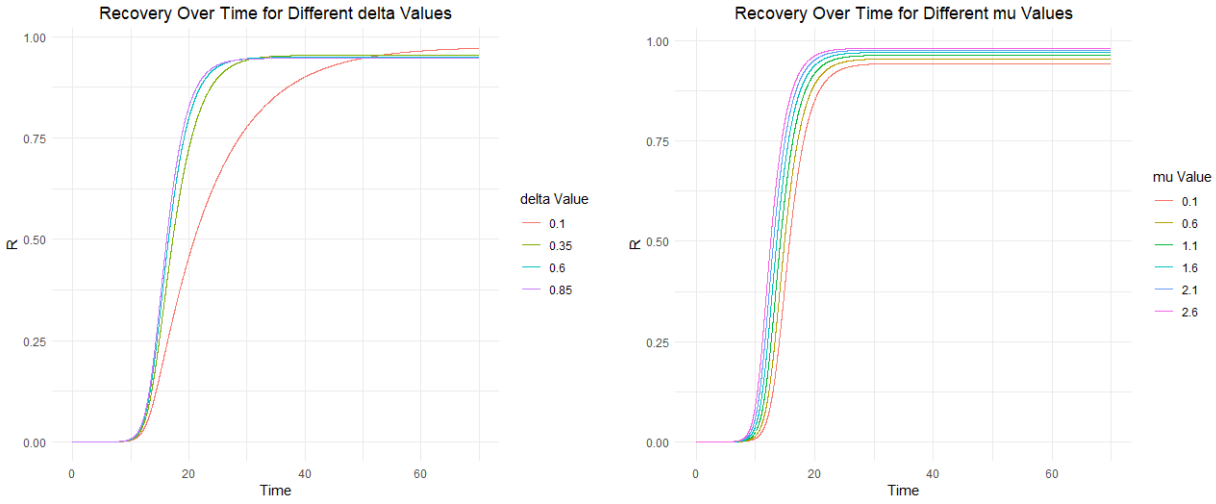

Figure 3: Recovery over time for varying parameters, δ (left) and μ (right) to illustrate the effect of the parameters on Recovery compartment.

Another observation is that R_0, the basic reproductive number representing the average number of secondary infections caused by a single infected individual in a fully susceptible population in our model can still be estimated by $R_0 = \frac{\beta}{\gamma}$, same as the original SIR model as Treated compartment is 0 at time 0. Therefore, the treated compartment has no effect on R_0 although setting higher δ and lower μ can lead to equilibrium being reached faster.

Discussion

It is difficult to check whether this model can provide an accurate description of reality. One of the ways to do this would be to compare the output of the model to real-world data collected under similar conditions to the model assumptions and adjust model's parameters to match that of the real-world conditions.

To further understand epidemic dynamics, further steady state analysis can also be done. For example, equilibrium analysis by setting system time derivatives to 0. In the case of our closed model, this would mean the disease has died out. On top of this stability analysis can also be done using Jacobian matrix to see the effect of reintroduction of a small number of infected individual(s) to the population. (Rafiq et al., 2022)

It is also possible to increase the complexity of the model further, for example, to include birth and death, migration, heterogeneity of the population, heterogeneity of interaction and so on. However, the decision to increase complexity should depend on the goal of the model and what it is attempting to represent as increasing the complexity of the model can lead to difficulties in variables and parameters estimation, worst generality, and increased uncertainty. (Puy et al., 2022)

Conclusion

By adding an extra compartment to SIR model, we increased model complexity and was able to draw conclusions that having a portion of infected individuals undergoing treatments during an epidemic can accelerate the end of the pandemic depending on the rate of individuals starting treatment, the rate of recovery, the rate of infection etc., and possibly prevent a portion of susceptible individuals in the population from ever being infected by the disease.

```
# loading libraries
library(deSolve)
library(reshape2)
library(ggplot2)

# defining sir function:
# β: beta infection rate for untreated individuals
# δ: delta recovery rate for treated infected individuals
# ϵ: epsilon rate at which treated individuals revert to the infected compartment
# γ: gamma recovery rate for untreated infected individuals
# μ: mu infection rate for treated individuals
# θ: theta treatment initiation rate for infected individuals

sir <- function(time, state, parameters) {
  with(as.list(c(state, parameters)), {
    dS <- -(beta * I + mu * T) * S
    dI <- (beta * I + mu * T)* S - (gamma + theta) * I + epsilon * T
    dT <- theta * I - (epsilon + delta) * T
    dR <- gamma * I + delta * T
    return(list(c(dS, dI, dR, dT)))
  })
}

# initial proportions in each compartment
init <- c(S = 0.999999,
          I = 0.000001,
          R = 0,
          T = 0)

# set parameters
parameters <- c(beta = 1.5,
                delta = 2,
                epsilon = 0,
                gamma = 0.1,
                mu = 0.1,
                theta = 0.4
                )

# set time frame
times <- seq(0, 70, by = 0.01)

# solve using ode
out <- ode(y = init, times = times, func = sir, parms = parameters)

# convert to a data frame
out <- as.data.frame(out)
head(out)
out_long <- melt(out, id.vars = "time", measure.vars = c("S", "I", "T", "R"),
                 variable.name = "Compartment", value.name = "value")
names(out_long) = c("time","Compartment","value")

################################
####part 1: plotting graph####
################################

ggplot(data = out_long, aes(x = time, y = value / max(value))) +
  geom_line(aes(col = Compartment, group = Compartment)
            #, size = 1.2
            ) +
  labs(title = "SITR Model") +
  xlab("Time") +
  ylab("Proportion of Population") +
  scale_color_manual(values = c("#00FF00", "#FF0000", "#56B4E9", "#A020F0")) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5)) +
  ylim(0, 1)

#############################################################
####part 2: varying parameters to check for its effects####
#############################################################

plot_list <- list()

parameters <- c(beta = 1.5,
                delta = 2,
                epsilon = 0,
                gamma = 0.1,
                mu = 0.1,
                theta = 0.4
)

varied_parameter <- "delta"
# values used: "delta"  seq(0.1, 1, by = 0.25)

# varied_parameter <- "mu"
# values used: "mu"  seq(0.1, 1.5, by = 0.1)

# Loop through different values of varied_parameter
for (val in  seq(0.1, 1, by = 0.25)) {
  # Update the delta parameter
  parameters[varied_parameter] <- val

  # Solve the differential equations
  out <- ode(y = init, times = seq(0, 70, by = 0.01), func = sir, parms = parameters)
```

```
  # Convert to a data frame
  out <- as.data.frame(out)

  out_long <- melt(out, id.vars = "time", measure.vars = c("S", "I", "T", "R"),
                   variable.name = "Compartment", value.name = "value")

  names(out_long) <- c("time", "Compartment", "value")

  # Plotting graph
  plot_obj <- ggplot(data = out_long, aes(x = time, y = value / max(value))) +
    geom_line(aes(col = Compartment, group = Compartment)#, size = 1.2
              ) + labs(title = paste(varied_parameter, val,"SITR Model")) +
    xlab("Time") +
    ylab("Proportion of Population") +
    scale_color_manual(values = c("#00FF00", "#FF0000", "#56B4E9", "#A020F0")) +
    theme_minimal() +
    theme(#text = element_text(size = 20),
          plot.title = element_text(hjust = 0.5)) +
    ylim(0, 1)

  # Store the plot object in the list
  plot_list[[as.character(val)]] <- plot_obj
}

# Display the plots
for (i in seq_along(plot_list)) {
  print(plot_list[[i]])
}

# to delete all the plots:
# dev.off(dev.list()["RStudioGD"])

#########################################################################################################################################
####part 3: plotting Recover for different parameter to visualize the effect of changes in parameter to the model dynamic####
#########################################################################################################################################

# Function to vary parameters (with initial values hard coded into the function)
vary_parameter <- function(param_name, param_values, init, times) {
  results <- data.frame()

  for (val in param_values) {
    parameters <- c(beta = 1.5, gamma = 0.1, delta = 2, theta = 0.4, epsilon = 0, mu = 0.1) # Initial parameters for the function
    parameters[param_name] <- val

    out <- ode(y = init, times = times, func = sir, parms = parameters)
    out_long <- melt(as.data.frame(out), id.vars = "time", measure.vars = c("S", "I", "T", "R"),
                     variable.name = "Compartment", value.name = "value")
    out_long[[param_name]] <- val
    results <- rbind(results, out_long)
  }

  return(results)
}

# Initial proportions in each compartment
init <- c(S = 0.999999, I = 0.000001, R = 0, T = 0)

# Time frame
times <- seq(0, 70, by = 0.01)

# Varying parameter and its sequence
param_name <- "mu"  # parameter to vary
param_values <- seq(0.1, 2.6, by = 0.5)  # set sequence to vary the parameter by

# param_name <- "delta"  # parameter to vary
# param_values <- seq(0.1, 1, by = 0.25)  # set sequence to vary the parameter by

# Generate results for varying parameter
results <- vary_parameter(param_name, param_values, init, times)

# Plotting
ggplot(results[results$Compartment == "R",], aes(x = time, y = value, color = as.factor(.data[[param_name]]))) +
  geom_line() +
  labs(x = "Time", y = "R", color = paste(param_name, "Value")) +
  theme_minimal() +
  ggtitle(paste("Recovery Over Time for Different", param_name, "Values")) +
  theme(plot.title = element_text(hjust = 0.5))


#############################
####part 4: phase diagram####
#############################

# Plotting Infected (I) against Susceptible (S)
ggplot(data = out, aes(x = S, y = I, color = time)) +
  geom_point() +
  labs(title = "Phase Diagram") +
  xlab("Susceptible (S)") +
  ylab("Infected (I)") +
  scale_color_gradientn(colors = c("blue", "green", "yellow", "red")) +
  theme(plot.title = element_text(hjust = 0.5)) +
  xlim(0, 1) +
  ylim(0, 1)
```

Reference list

Carvalho, A.M. and Sebastián Gonçalves (2021). An analytical solution for the Kermack–McKendrick model. Physica A: Statistical Mechanics and its Applications, [online] 566, p.125659. doi:https://doi.org/10.1016/j.physa.2020.125659.

Puy, A., Beneventano, P., Levin, S.A., Lo Piano, S., Portaluri, T. and Saltelli, A. (2022). Models with higher effective dimensions tend to produce more uncertain estimates. Science Advances, 8(42). doi:https://doi.org/10.1126/sciadv.abn9450.

Rafiq, M., Ali, J., Riaz, M.B. and Awrejcewicz, J. (2022). Numerical analysis of a bi-modal covid-19 SITR model. Alexandria Engineering Journal, 61(1), pp.227–235. doi:https://doi.org/10.1016/j.aej.2021.04.102.