Epidemiological Modelling: Course Notes

Bram Kuijper, Mario Recker

2024-02-27

Contents

Chapter 1

Lecture Notes from uncertainties to curve fitting

Here the lecture notes with hints to use during the tutorials and the questions for you to solve during the practical.

1.1

1.2 Copying code

If you hover over the top right corner of a code block in these lecture notes, a copy button emerges that allows you to copy the code (and then paste it into Rstudio) without having to first select all of the code.

Chapter 2

Tutorial uncertainties

2.1 Estimating R_0 in a risk-heterogenous population

In the first of this tutorial, we will run an SIS model of a population heterogenous for risk. We then estimate R_0 for different WAIFW matrices (Who Acquires Infection From Whom)

2.1.1 SIS model code

Inspect the model code below (particularly the set of differential equations) and compare the equations with the flow diagram of the SIS model in the lecture slides.

```
library("tidyverse")
library("deSolve")

# set of differential equations
# of a model in which classes vary
# in their transmission risk
SIS_structure <- function(t, y, parameters)
{
    # get state variables from
    # the y function argument
    SL = y[1]
    SH = y[2]
    IL = y[3]
    IH = y[4]

    # with() creates a 'sub environment'
```

```
    # in which all the named values of the
    # parameters list are now variables by themselves
    # hence, we can then evaluate all the gradients
    # within this environment
    result <- with(data=as.list(parameters),
    expr = {
        # evaluate each of the equations
        dSL <- -mu * SL - betaLL * SL * IL - betaHL * SL * IH + gamma * IL

        dSH <- -mu * SH - betaLH * SH * IL - betaHH * SH * IH + gamma * IH

        dIL <- betaHL * SL * IH + betaLL * SL * IL - (gamma  + mu) * IL

        dIH <- betaLH * SH * IL + betaHH * SH * IH - (gamma  + mu) * IH

        # then return the gradient values
        c(dSL,dSH,dIL,dIH)
    })

    # return list of gradients
    return(list(result))
} # end SIR_structure

# set out the time points
times <- seq(from = 0, to = 25, length.out = 1000)

# make a named vector containing all
# the parameters used in the model
params <- c(
    betaHH=5,
    betaHL=0.1,
    betaLH=0.1,
    betaLL=2,
    gamma=1,
    mu=0,
    N=1
)

# a set of starting values in which we
# assume that equally frequent
start <- c(SL=0.495,SH=0.495,IL=0.005,IH=0.005)

# run the model and obtain a data.frame
# with densities over time
output <- as.data.frame(ode(y = start
```

```
              ,times = times
              ,func=SIS_structure
              ,parms = params))
```

2.1.2 Task: estimate R0 without taking into account risk

After having inspected the model code, try to run it. There should now be an `output` variable, which contains a `data.frame`. This `data.frame` contains the the time evolution of the densities of susceptibles and infecteds.

Try to get the total proportion of susceptibles from the `output data.frame`, by summing the low and high risk susceptibles. Use this number as your value of p_S and then calculate R_0 from that (see the lecture slides for the corresponding formula). *Hint*: you need to use the equilibrium values of the number of susceptibles.

2.1.3 Task: estimate R0 while taking into account risk

Now we will try to obtain a more precise measure of R_0, by using the WAIFW matrix and the equilibrium values of S_H and S_L which you obtained in the previous subsection.

Use R's `matrix()` command to fill out the following 2×2 matrix, using the densities S_H, S_L and the parameter values of **beta** that you used to run the SIS model

$$\mathbf{R} = \begin{bmatrix} \beta_{HH} S_H, \beta_{HL} S_L \\ \beta_{LH} S_L, \beta_{LL} S_L \end{bmatrix}$$

Then use the `eigen()` command on this matrix to calculate the dominant (i.e., the largest) eigenvalue, which is R_0 as it calculates the overall number of secondary cases, while taking into account the different contributions of high and low risk individuals.

Associated to the dominant eigenvalue is the dominant eigenvector, absolute values of which inform one about the relative long-term contribution in spreading the epidemic by high versus low-risk individuals.

2.2 Maximum Likelihood and age-dependent seropositivity

For a disease with an unknown R_0, we provide three datasets with ages and whether individuals are seropositive or seronegative. We will then use a maximum likelihood approach to estimate the value of R_0 that results from this data.

2.2.1 Probabilities

Let $P(a_i) = \exp\left[-a_i\mu\left(R_0 - 1\right)\right]$ be the probability that the ith individual sampled of age a_i is still susceptible (i.e., has no antibodies). Similarly, $Q(a_i) = 1 - \exp\left[-a_i\mu\left(R_0 - 1\right)\right]$ reflects the probability

Chapter 3

Parts

You can add parts to organize one or more book chapters together. Parts can be inserted at the top of an .Rmd file, before the first-level chapter heading in that same file.

Add a numbered part: `# (PART) Act one {-}` (followed by `# A chapter`)

Add an unnumbered part: `# (PART\*) Act one {-}` (followed by `# A chapter`)

Add an appendix as a special kind of un-numbered part: `# (APPENDIX) Other stuff {-}` (followed by `# A chapter`). Chapters in an appendix are prepended with letters instead of numbers.

Chapter 4

Fitting dynamic models to data

4.1 Hong Kong flu dataset

Please find a dataset on the 1968 Hong Kong flu outbreak in New York below:

```
flu <- data.frame(week = 1:13,
                  deaths = c(14, 28, 50, 66, 156, 190, 156, 108, 68, 77, 33, 65, 24))
```

We will now use `optim()` to try and fit this dataset and estimate values for the transmission rate β and the disease clearance rate γ.

4.1.1 Setting up the SIR model

First, we code up a SIR ODE model with frequency-dependent transmission (as in the previous practicals) that can be solved by `deSolve`'s `ode()` function. Again, we ignore demography similar to the SIR model presented on day 1. For example, this would be the skeleton of such a function:

```
sir_ode <- function(t, demographic_variables, parameters)
{
    with(as.list(demographic_variables, parameters)
         {
           dS <- -beta * S * I /(S + I + R)
    })
}
```

4.2 Task: finalize the `sir_ode` code above

Please try to finalize the `sir_ode` code above taking care that the function should return a list of gradients, i.e., $\frac{\mathrm{d}S}{\mathrm{d}t}, \frac{\mathrm{d}I}{\mathrm{d}t}, \frac{\mathrm{d}R}{\mathrm{d}t}$

4.3 Interfacing the model with the data

The most important part of this tutorial is to make a *goal function* which attaches a value to how well the model fits the data.

We assume that the number of deaths from the flu data set above is equal to the number of infected individuals produced by the ODE model

To this end, the goal function should receive the following: (i) the current guess of the parameters β and γ (for which it will calculate some measure of goodness-of-fit, such as the sum of squares, SS) (ii) some information about initial densities and (iii) the actual data of the course of the infection. Let's do this:

```
library("deSolve")
goal.function <- function(parameters, initial_densities, data)
{
  # first obtain the time points from the dataset
  times_from_data <- data[,"week"]

  # solve the SIR ODE over time.
  # store the result as a data.frame
  the.ode.data <- as.data.frame(ode(
    y = initial_densities,
    times = times_from_data,
    func = sir_ode, parms = parameters))

  # now extract the numbers of infecteds from
  # the resulting ode data and compare to
  # the ODE
  # and calculate the sum of squares between
  # the data and the ODE
}
```

4.4 Task: try to finish the function code above

Try to finalize the final bit of the `goal.function` code above, by calculating the sum of squares between the number of deaths in the `data` and the density of infecteds resulting from the ODE, contained in `the.ode.data`. *Hint* use R's `sum()` function to sum over the squared difference between the data and the ODE.

4.5 Task: test driving our function

We can then test-drive our `goal.function()` by simply running it with a bunch of arguments that we made up and see whether there are any errors.

```
# lucky guess of beta and gamma resulting in
# an R0 of 1.25
# we need to provide this as a name-value vector
pars <- c(beta = 1, gamma = 0.8)
# lets assume a large number of susceptibles
# but we could potentially vary this later
init_dens <- c(S=1e05,I=1,R=0)

goal.function(parameters = parameters
              ,initial_densities = initial_densities
              ,data = flu
              )
```

Now that we have our goal function in hand we have to feed the goal function to `optim()` which helps us find the lowest sum of squares

```
optim_result <- optim(
  par = pars
  ,fn = goal.function
  ,initial_densities = init_dens
)
```

4.6 Task:

4.7 Task: missing data

Let's imagine that local health services have been slow to pick up on the disease, so that the last four data points are missing. Hence the data set is now as follows:

```
flu <- data.frame(week = 1:13,
                  deaths = c(14, 28, 50, 66, 156, 190, 156, 108, 68, 77, 33, 65, 24))
```

Chapter 5

Footnotes and citations

5.1 Footnotes

Footnotes are put inside the square brackets after a caret `^[]`. Like this one [1].

5.2 Citations

Reference items in your bibliography file(s) using `@key`.

For example, we are using the **bookdown** package [**?**] (check out the last code chunk in index.Rmd to see how this citation key was added) in this sample book, which was built on top of R Markdown and **knitr** [**?**] (this citation was added manually in an external file book.bib). Note that the `.bib` files need to be listed in the index.Rmd with the YAML `bibliography` key.

The RStudio Visual Markdown Editor can also make it easier to insert citations: https://rstudio.github.io/visual-markdown-editing/#/citations

[1] This is a footnote.

Chapter 6

Blocks

6.1 Equations

Here is an equation.

$$f(k) = \binom{n}{k} p^k (1-p)^{n-k} \tag{6.1}$$

You may refer to using `\@ref(eq:binom)`, like see Equation (6.1).

6.2 Theorems and proofs

Labeled theorems can be referenced in text using `\@ref(thm:tri)`, for example, check out this smart theorem 6.1.

Theorem 6.1. *For a right triangle, if c denotes the* length *of the hypotenuse and a and b denote the lengths of the* ***other*** *two sides, we have*

$$a^2 + b^2 = c^2$$

Read more here https://bookdown.org/yihui/bookdown/markdown-extensions-by-bookdown.html.

6.3 Callout blocks

The R Markdown Cookbook provides more help on how to use custom blocks to design your own callouts: https://bookdown.org/yihui/rmarkdown-cookbook/custom-blocks.html

Chapter 7

Sharing your book

7.1 Publishing

HTML books can be published online, see: https://bookdown.org/yihui/bookdown/publishing.html

7.2 404 pages

By default, users will be directed to a 404 page if they try to access a webpage that cannot be found. If you'd like to customize your 404 page instead of using the default, you may add either a `_404.Rmd` or `_404.md` file to your project root and use code and/or Markdown syntax.

7.3 Metadata for sharing

Bookdown HTML books will provide HTML metadata for social sharing on platforms like Twitter, Facebook, and LinkedIn, using information you provide in the `index.Rmd` YAML. To setup, set the `url` for your book and the path to your `cover-image` file. Your book's `title` and `description` are also used.

This `gitbook` uses the same social sharing data across all chapters in your book- all links shared will look the same.

Specify your book's source repository on GitHub using the `edit` key under the configuration options in the `_output.yml` file, which allows users to suggest an edit by linking to a chapter's source file.

Read more about the features of this output format here:

https://pkgs.rstudio.com/bookdown/reference/gitbook.html

Or use:

```
?bookdown::gitbook
```

Bibliography