Simulation-based model selection for population biology - STAT 610 - (paper results replication)

Trang Nguyen

Contents

Context :

In this project, I will replicate the results from the paper "Simulation-based model selection for population biology" by Toni et al. (2009). The goal is to implement the simulation-based model selection approach described in the paper and apply it to a population biology scenario.

The scenario considered is about the spread of different strains of the influenza virus. The data used in the paper comes from influenza A (H3N2) outbreaks that occurred in 1977-1978 and 1980-1981 in Tecomseh, Michigan, (Supplementary table 2) and a second dataset of an influenza B infection outbreak in 1975-1976 and influenza A (H1N1) infection outbreak in 1978-1979 in Seattle, Washington (Supplementary table 3).

Questions:

In the paper, the authors focused on two main questions regarding the influenza outbreaks. 1. Can different outbreaks of the same strain be described by the same model of disease spread? 2. Can outbreaks of different strains be described by the same model of disease spread?

Assumptions of the paper :

In the paper, the authors assumed that : - The virus can be spread from the Infected individuals to the Susceptible individuals. - The spread can occur both within households and across the population at large.

Parameters to infer :

- q_c : the probability that a susceptible individual does not get infected from the community.
- q_h : the probability that a susceptible individual escapes infection within their household.
- w_{js} : the probability that j out of the s susceptibles in a household become infected,

$$w_{js} = \binom{s}{j} w_{jj} (q_c q_h^j)^{s-j}$$

where $w_{0s} = q_{cs}, s = 0, 1, 2, \dots$ and $w_{jj} = 1 - \sum_{i=0}^{j-1} w_{ij}$.

We want to infer q_c and q_h using data from Supplementary table 2 and table 3. In this paper, they considered two models : one with four parameters (one pair for each outbreak) and one with two parameters (shared between the two outbreaks).

Predefined parameters :

- Population size N = 1000
- Prior distributions of all parameters are chosen to be uniform over the range [0, 1].
- The distance function used to compare simulated and observed data is defined as follows :

$$d(D_0, D^*) = \frac{1}{2}||D_1 - D^*(q_{h1}, q_{c1})||_F + \frac{1}{2}||D_2 - D^*(q_{h2}, q_{c2})||_F$$

$D_0 = D_1 \cup D_2$ is the observed data (the combination of the two outbreaks D_1 and D_2) and D^* is the simulated data from the model. The Frobenius norm is used to measure the difference between the two datasets.

3. Summaries for Supplementary Table 2 (for Fig. 3a descriptive stats)

```r
t_last = length(res_suppl2$epsilon)

# Posterior particles with named parameters
post_df_suppl2 = get_posterior_particles(res_suppl2, t = t_last)
post_df_suppl2 = label_parameters_suppl2(post_df_suppl2)

# Posterior model probabilities (this is Fig. 3b numerically)
posterior_model_probs_fun(res_suppl2)
```

```
##   model       prob
## 1     1 0.97311203
## 2     2 0.02688797
```

```r
# Summaries for the four-parameter model (model 2)
post4 = subset(post_df_suppl2, model == 2)

params = c("q_c1", "q_h1", "q_c2", "q_h2")
summaries = lapply(params, function(p) {
  x = post4[[p]]
  w = post4$weight
  c(mean = sum(x * w) / sum(w),
    weighted_quantile(x, w))
})
names(summaries) = params
summaries
```

```
## $q_c1
##      mean
## 0.8674782 0.8441640 0.8669045 0.8948911
##
## $q_h1
##      mean
## 0.8479023 0.7616150 0.8467207 0.9320475
##
## $q_c2
##      mean
## 0.8837344 0.8578765 0.8834295 0.9146760
##
## $q_h2
##      mean
## 0.7935169 0.6791663 0.8030042 0.9368755
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.4.2
```

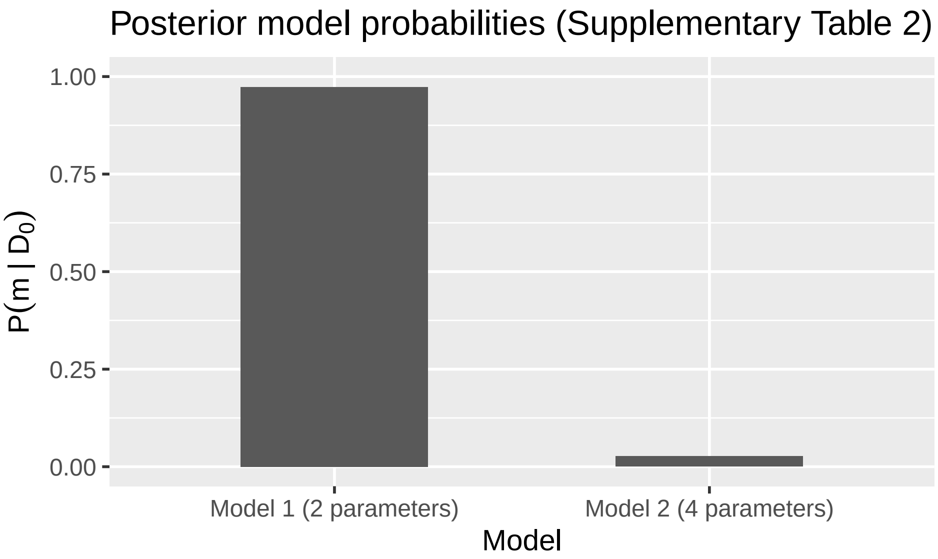
```
library(dplyr)
library(tidyr)
```

```
# Restrict to four-parameter model (model 2)
post4 = subset(post_df_suppl2, model == 2)

# Resample from weighted particles to get an unweighted sample
set.seed(123)
n_sample = 10
idx = sample(seq_len(nrow(post4)), size = n_sample,
              replace = TRUE, prob = post4$weight)
samp4 = post4[idx, ]

# Long format: q_c and q_h, for each outbreak
qc_long = rbind(
  data.frame(outbreak = "1977-1978", param = "q_c", value = samp4$q_c1),
  data.frame(outbreak = "1980-1981", param = "q_c", value = samp4$q_c2)
)

qh_long = rbind(
  data.frame(outbreak = "1977-1978", param = "q_h", value = samp4$q_h1),
  data.frame(outbreak = "1980-1981", param = "q_h", value = samp4$q_h2)
)

# Posterior of q_c (panel corresponding to Fig. 3a, first row)
p_qc = ggplot(qc_long, aes(x = value, colour = outbreak, fill = outbreak)) +
  geom_density(alpha = 0.2) +
```

```r
  labs(x = expression(q[c]), y = "Posterior density",
       title = "Posterior of q[c] (four-parameter model, Supplementary Table 2)") +
  xlim(0, 1)

# Posterior of q_h (panel corresponding to Fig. 3a, second row)
p_qh = ggplot(qh_long, aes(x = value, colour = outbreak, fill = outbreak)) +
  geom_density(alpha = 0.2) +
  labs(x = expression(q[h]), y = "Posterior density",
       title = "Posterior of q[h] (four-parameter model, Supplementary Table 2)") +
  xlim(0, 1)
```

5. Reproducing Fig. 3b (posterior model probabilities $P(m \mid D_0)$)

```r
post_model_probs2 = posterior_model_probs_fun(res_suppl2)

post_model_probs2$model = factor(
  post_model_probs2$model,
  levels = c(1, 2),
  labels = c("Model 1 (2 parameters)", "Model 2 (4 parameters)")
)

ggplot(post_model_probs2, aes(x = model, y = prob)) +
  geom_col(width = 0.5) +
  ylim(0, 1) +
  labs(x = "Model",
       y = expression(P(m ~ "|" ~ D[0])),
       title = "Posterior model probabilities (Supplementary Table 2)")
```



This is the analogue of Fig. 3b: a bar for the 2-parameter model vs the 4-parameter model. Toni & Stumpf report $\Pr(m = 1 \mid D_0) \approx 0.98$ as a median over 10 runs; your single run will be in that neighbourhood but not exactly identical.

6. How to extend to Fig. 3c-d (Seattle outbreaks)

Once you:

- run an analogous ABC-SMC for **Supplementary Table 3** (either:
  - four-parameter model only, to get Fig. 3c, and
  - model 1 = three parameters (q_{c1}, q_{c2}, q_h) vs model 2 = four parameters, for Fig. 3d),

you can reuse **exactly** the same utilities:

- `get_posterior_particles()` → label parameters appropriately (you'll want a `label_parameters_suppl3()` for the 3-parameter/4-parameter semantics).
- `posterior_model_probs_fun()` → bar plot for Fig. 3d.
- The density plotting pattern from `fig3a-replot` → two density panels for q_c and q_h for the four-parameter model (Fig. 3c).

But for the current code you pasted (set up for Suppl. Table 2, 2- vs 4-parameter models), the chunks above give you:

- numerical posterior summaries,
- a clean re-plot of Fig. 3a (for q_c and q_h), and
- a re-plot of Fig. 3b (posterior model probs).