State-transition model dynamic - Sick-Sicker case study

Eline Krijkamp, MSc[†]
Fernando Alarid-Escudero[1], PhD[‡]
Eva A. Enns, PhD[3]
Myriam G.M. Hunink, PhD[2][§]
Petros Pechlivanoglou, PhD[¶]
Hawre Jalal, MD, PhD[‖]

2019-02-12

Sick-Sicker model

In the Sick-Sicker model, we simulate a hypothetical cohort of 25-year-old individuals over a lifetime (or reaching age 100-years old) using 75 annual cycles, represented with `n.t`. The cohort start in the "Healthy" health state (denoted H). Healthy individuals are at risk of developing the illness, at which point they would transition to the first stage of the disease (the "Sick" health state, denoted S1). Individuals that become sick incur a one-time utility decrement of 0.01 (`du.HS1`), disutility of transitioning from H to S1) and a one-time cost of $1,000 (`ic.HS1`) that reflect the acute impacts of developing the illness. Sick individuals are at risk of further progressing to a more severe stage (the Sicker health state, denoted S2), which is constant in this case example. There is a chance that individuals in the Sick state eventually recover and return back to the Healthy state. However, once an individual reaches the Sicker health state, they cannot recover; that is, the probability of transitioning to the "Sick" or "Healthy" health states from the Sicker health state is zero. Individuals in the "Healthy" state face background mortality that is age-specific (i.e., time-dependent). Sick and Sicker individuals face an increased mortality in the form of a hazard rate ratio (HR) of 3 and 10 times, respectively, on the background mortality rate. Sick and Sicker individuals also experience increased health care costs and reduced QoL compared to healthy individuals. Once simulated individuals die, they transition to the Dead health state (denoted D), where they remain. When an individual dies, they incur a one-time cost of $2,000 (`ic.D`) that reflects the acute care that might be received immediately preceding death. The state-transition diagram of the Sick-Sicker model is shown in Figure 1. The evolution of the cohort is simulated in one-year discrete-time cycles. Both costs and QALYs are discounted at an annual rate of 3%.

01 Initial setup

02 Define and initialize matrices and vectors

Equation 1

```
#### 02.1 Transition probability matrix ####
# the m.P transition probability matrix at the first cycle
f.create_transition_prob_matrix(v.params = v.params.init, t = 1)
```

```
##           H         S1        S2          D
## H  0.8489865 0.1500000 0.0000000 0.001013486
## S1 0.5000000 0.3919626 0.1050000 0.003037378
```

[*]Equal contribution to this paper
[†]Department of Epidemiology, Erasmus MC, University Medical Center, Rotterdam, The Netherlands
[‡]Division of Health Policy and Management, University of Minnesota School of Public Health, Minneapolis, MN, USA
[§]Center for Health Decision Sciences, Harvard T.H. Chan School of Public Health, Boston, USA
[¶]The Hospital for Sick Children, Toronto and University of Toronto, Toronto, Ontario, Canada
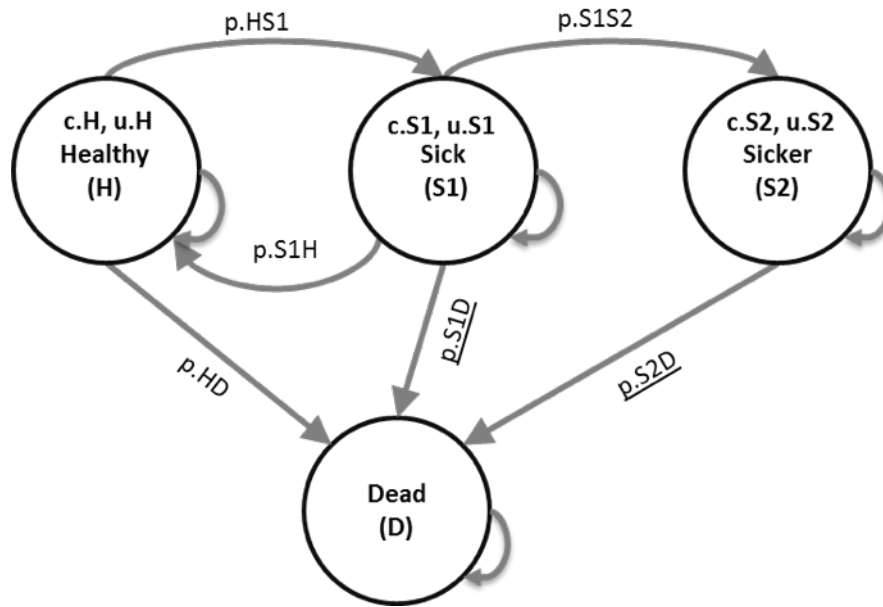[‖]University of Pittsburgh, Pittsburgh, PA, USA

Figure 1: Sick-Sicker

```
## S2 0.0000000 0.0000000 0.9899112 0.010088764
## D  0.0000000 0.0000000 0.0000000 1.000000000
```

```
#### 02.2 Initial state vector ####
# the cohort start in the Healthy health state
s0 <- c(H = 1, S1 = 0, S2 = 0, D = 0)
s0
```

```
##  H S1 S2  D
##  1  0  0  0
```

```
#### 02.3 Cohort trace
## Create the Markov trace matrix M capturing the proportion of the cohort
# in each state at each cycle

m.M <- matrix(0,  # initialize cohort trace
              nrow = (n.t + 1), ncol = n.states,
              dimnames = list(0:n.t, v.n))

m.M[1, ] <- s0   # store the initial state vector
```

03 Matrix Approach

Equation 2

```
for(t in 1:n.t){  # loop through the number of cycles
  # create the transition probability matrix for the current cycle
  m.P <- f.create_transition_prob_matrix(v.params = v.params.init, t = t)
  # estimate the state vector for the next cycle (t + 1)
  m.M[t + 1, ] <- m.M[t, ] %*% m.P   # Equation 2
}
```

```r
head(round(m.M, 3)) # show the first six lines of the Markov trace
```

```
##       H    S1    S2    D
## 0 1.000 0.000 0.000 0.000
## 1 0.849 0.150 0.000 0.001
## 2 0.796 0.186 0.016 0.002
## 3 0.769 0.192 0.035 0.004
## 4 0.749 0.191 0.055 0.006
## 5 0.731 0.187 0.074 0.008
```

04 Array Approach

```r
a.A <- array(0, dim = c(n.states, n.states, n.t + 1),
             dimnames = list(v.n, v.n, 0:n.t)) # initialize array
diag(a.A[, , 1]) <- s0 # store the initial state vector in the diagnal of A
```

Equation 3 & 4

```r
a.A[, , 1]
```

```
##    H S1 S2 D
## H  1  0  0 0
## S1 0  0  0 0
## S2 0  0  0 0
## D  0  0  0 0
```

```r
# run the model
for(t in 1:n.t){                            # loop through the number of cycles
  # create the transition probability matrix for the current cycle
  m.P <- f.create_transition_prob_matrix(v.params = v.params.init, t = t)
#### Equation 4    ####
  a.A[, , t + 1] <- colSums(a.A[, , t]) * m.P  # fill array A for t + 1
}
```

Equation 5

```r
a.A[, , 2:3] # shown for two cycles
```

```
## , , 1
##
##            H   S1 S2           D
## H  0.8489865 0.15  0 0.001013486
## S1 0.0000000 0.00  0 0.000000000
## S2 0.0000000 0.00  0 0.000000000
## D  0.0000000 0.00  0 0.000000000
##
## , , 2
##
##            H        S1      S2           D
## H  0.7207908 0.12734798 0.00000 0.000847714
## S1 0.0750000 0.05880112 0.01575 0.000448877
## S2 0.0000000 0.00000000 0.00000 0.000000000
## D  0.0000000 0.00000000 0.00000 0.001013486
```

Equation 7

```r
# calculating M from A
m.M_A <- t(colSums(a.A))    # sum over the colums of A and transpose
```

05 Apply state and transtion rewards

```r
#### 05.1 Create reward matrices for both costs and effects ####
m.R_costs   <- f.create_transition_reward_matrix_costs(v.params = v.params.init)
m.R_effects <- f.create_transition_reward_matrix_effects(v.params = v.params.init)
```

Equation 8

```r
m.R_costs     # show the reward matrix for costs
```

```
##         H     S1     S2     D
## H    2000  3000  2000  4000
## S1   4000  4000  4000  6000
## S2  15000 15000 15000 17000
## D       0     0     0     0
```

```r
m.R_effects   # show the reward matrix for effects
```

```
##        H   S1   S2    D
## H   1.00 0.99 1.00 1.00
## S1  0.75 0.75 0.75 0.75
## S2  0.50 0.50 0.50 0.50
## D   0.00 0.00 0.00 0.00
```

Equation 9

```r
#### 05.2 Expected QALYs and Costs per cycle for each strategy ####
a.O_costs <- a.O_effects <- array(0, dim = c(n.states, n.states, n.t + 1),
            dimnames = list(v.n, v.n, 0:n.t))

for(t in 1:n.t){
# element-wise-multiplication of array A with the rewards matrices
a.O_costs[, , t]   <- a.A[, , t] * m.R_costs
a.O_effects[, , t] <- a.A[, , t] * m.R_effects
}
```

Equation 10

```r
# calculate the expected costs per cycle
v.Costs <- rowSums(t(colSums(a.O_costs)))
# calculate the expected QALYs per cycle
v.QALYs <- rowSums(t(colSums(a.O_effects)))

TC <- t(v.Costs) %*% v.dwc  # calculate the total expected costs
TE <- t(v.QALYs) %*% v.dwe   # calculate the total expected QALYS

v.Results <- c(TC, TE)       # combine the total expected costs and QALYs
names(v.Results) <- c("Costs", "Effect") # name the vector
v.Results                    # print the results
```
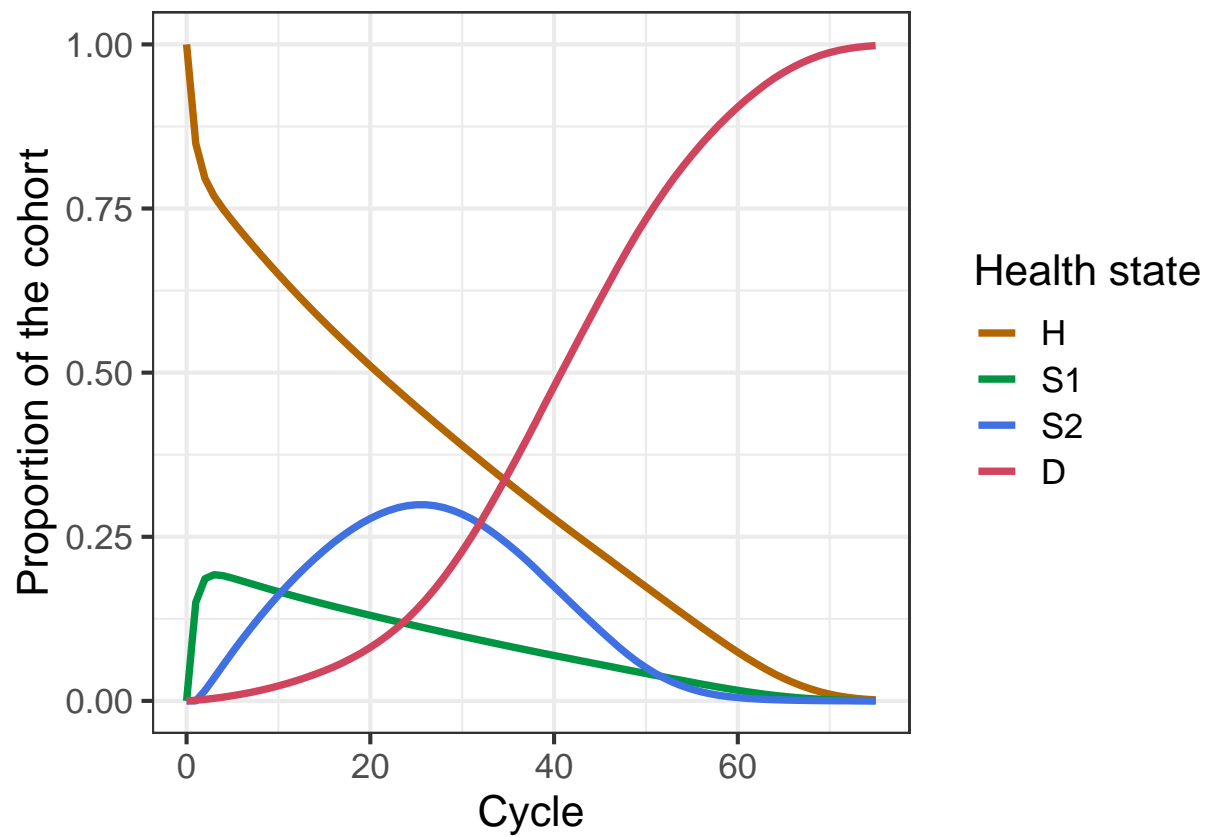
Figure 2: Cohort trace of the Sick-Sicker cohort model

```
##         Costs       Effect
## 115104.13376     20.37685
```

06 Plot cohort trace