8

Markov models — SOLUTIONS

8.1 Revision of beta+binomial conjugate Bayesian inference

This question can be omitted if you are confident about the material on conjugate Bayesian inference from lecture 2. However, it may help in understanding later questions on Markov models constructed from Dirichlet+Multinomial models, which are a generalisation of Beta+Binomial models.

a. Write down the likelihood function for $r = 15$ people falling asleep during the lecture hour, given $n = 40$ were awake at the start. (Assume these are independent events and everyone has the same probability, π, of falling asleep).
Solution: *The number r falling asleep follows a Binomial distribution with denominator n and probability π, with probability mass $p(r) = \binom{n}{r}\pi^r(1-\pi)^{n-r}$. So with $n = 40$ and $r = 15$, the likelihood function for π is $f(\pi) = \binom{40}{15}\pi^{15}(1-\pi)^{25}$.*

b. Assuming a Beta(1, 1) prior distribution for π, write down the posterior distribution, given the numbers in each state in the lecture hour.
Solution: *Recalling the second lecture, the posterior for a binary outcome probability π, given a Beta(a, b) prior and observed data r outcomes out of a denominator n, is Beta$(a + r, b + n - r)$. A Beta$(1, 1)$ is a uniform distribution. So in this example the posterior is a Beta$(16, 26)$ distribution. No need to write down the probability density function of this distribution.*

c. Simulate a vector of 1000 samples from this posterior distribution using Monte Carlo simulation in R
Solution: *Can be done in a single line of* R *code*

```
p <- rbeta(1000, 16, 26)
```

d. Assuming a time-homogeneous 2-state Markov chain with sleep as an absorbing state (people don't wake up again), obtain a vector of 1000 samples from the predictive distribution for the number awake after two hours.
Solution: *First note that the probability of being awake after one hour is $(1 - \pi)$, so the probability of staying awake for two hours is $(1 - \pi)^2$. Then the number awake after 2 hours can be drawn from a Binomial distribution with this probability, and denominator 40. Note that π is not fixed, but uncertain, as represented through its posterior distribution. We can sample a vector of 1000 values from the required predictive distribution in* R *as follows, using the random vector we drew in part (c). This distribution represents both uncertainty about the value of π, and random variability of outcomes given a fixed value of π.*

```
x <- rbinom(1000, 40, (1-p)^2)
```

Note in part (d) we have defined a simple Markov model with two states (awake→asleep), one allowed transition, two cycles, and transition probability matrix

$$\begin{pmatrix} (1-\pi) & \pi \\ 0 & 0 \end{pmatrix}$$

8.2 Dirichlet / Multinomial conjugate Bayesian inference

In the asthma example from the lecture, suppose that the model was simplified so that the two "exacerbation" states (Hex and Pex, hospital or primary-care managed exacerbations) are merged and considered as a single state (Ex: exacerbation).

a. Using data from the SFC treatment arm of the asthma trial in the lecture, write down the likelihood function for the transition probabilities out of state STW, given that $\boldsymbol{r} = (210, 60, 1, 1)$ people end up in states (STW, UTW, Ex, and TF) one week after being in state STW.
Solution: *The outcome* $\boldsymbol{r} = (r_1, r_2, r_3, r_4)$ *has a Multinomial distribution with probability vector* $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \lambda_3, \lambda_4)$. *and probability mass* $\propto \prod_{i=1}^{4} \lambda_i^{r_i}$. *Given these data, the likelihood is then proportional to* $\lambda_1^{210}\lambda_2^{60}\lambda_3\lambda_4$.

b. Assume a Dirichlet(1, 1, 1, 1) prior distribution for $\boldsymbol{\pi}_1$, the vector of probabilities for moving from state STW to the other states in one week. Write down the posterior distribution of $\boldsymbol{\pi}_1$, given data.
Solution: *The posterior given a Dirichlet*$(a_1, \ldots, a_4)$ *prior is Dirichlet*$(a_1 + r_1, \ldots, a_4 + r_4)$*, in this case, Dirichlet*$(211, 61, 2, 2)$.

c. Simulate a vector of 1000 samples from this posterior distribution using Monte Carlo simulation in R
Solution: *This can either be done through the gamma distribution in base* R

```
a <- c(211, 61, 2, 2)
p_rep <- matrix(nrow=1000, ncol=4)
for (i in 1:4) {
  p_rep[,i] <- rgamma(1000, a[i], 1)
}
p_rep <- p_rep / rowSums(p_rep)
p_rep[1:3,]
```

```
          [,1]      [,2]        [,3]        [,4]
[1,] 0.7119365 0.2460785 0.040800723 0.001184292
[2,] 0.7835474 0.2057707 0.004850106 0.005831843
[3,] 0.7903415 0.2038019 0.003278150 0.002578424
```

. . . or through a function in one of many add-on packages, e.g.

```
a <- c(211, 61, 2, 2)
# install.packages("VGAM") # if not already installed
library(VGAM)
p_rep <- rdiric(1000, a)
p_rep[1:3,]
```

```
          [,1]      [,2]        [,3]        [,4]
[1,] 0.7223305 0.2623305 0.009652661 0.005686361
[2,] 0.7068855 0.2740772 0.008836329 0.010200941
[3,] 0.8000553 0.1872717 0.008297979 0.004375051
```

8.3 Markov modelling in R

In this section, we will construct a Markov model from the asthma data given in the lecture, using the simplified four-state representation (STW, UTW, Ex, and TF).

a, Suppose that before this study, we have observed people visiting state Ex on about 100 occasions, but only once did somebody stay in that state for a period of more than one week. Also suppose we are unsure about what happens to people in the week following a period in the Ex state. Construct a Dirichlet prior for the transition probabilities out of state Ex based on this information.
Solution: *Dirichlet*$(33, 33, 1, 33)$ *or something close to this. The Dirichlet parameters* a_1, a_2, a_3, a_4 *should add up to about* $\sum_{i=1}^{4} a_i = 100$*, our prior sample size. The prior probabilility of staying in Ex for a second week has mean* $a_3 / \sum_{i=1}^{4} a_i = 1/100$*, so* $a_3 = 1$*. Then if we assume an equal prior probability that the next state is STW,UTW or TF, we have* $a_1 = a_2 = a_4 = 33$*.*

b. Suppose we observed the following transition count data

	to state			
from state	STW	UTW	Ex	TF
STW	210	60	1	1
UTW	88	641	4	13
Ex	1	0	0	1

Extend the code from section 8.2 to generate a sample from the posterior distribution of the full transition probability matrix, using the prior from part (a) for the transition probabilities from state Ex, and uniform Dirichlet priors for the transition probabilities out of STW and UTW, and noting that TF is an absorbing state. Thus estimate the posterior mean transition probability matrix.
Solution: *First define a matrix of the prior parameters for the Dirichlet distribution. Rows of all matrices here are defined by the starting state, columns by the destination state. Note that labelling the rows and columns is helpful, since it allows us to easiy refer to a needed row as, say,* `a["Ex",]`*, without needing to know that Ex is in row 3. The parameters for TF define a point mass on a probability of 1 for staying in TF in the following week.*

```
a <- matrix(nrow=4, ncol=4) # or array(dim=c(4,4))
state_names <- c("STW","UTW","Ex","TF")
rownames(a) <- colnames(a) <- state_names
a["STW",] <- c(1,1,1,1)
a["UTW",] <- c(1,1,1,1)
a["Ex",] <- c(33,33,1,33)
a["TF",] <- c(0,0,0,1)
```

Then define a matrix containing the observed transition counts, and finally obtain the posterior Dirichlet parameters by adding the prior parameters to the counts.

```
N <- rbind(
    "STW"=c(210,60,1,1),
    "UTW"=c(88,641,4,13),
    "Ex"=c(1,0,0,1),
    "TF"=c(0,0,0,0)
)
a_post <- a + N
```

We can then obtain a 3 dimensional array of samples from the posterior. This consists of 1000 random 4×4 *matrices.*

```
lambda_rep <- array(dim=c(1000, 4, 4))
dimnames(lambda_rep) <- list(NULL, state_names, state_names)
for (i in 1:4){
    lambda_rep[,i,] <- rdiric(1000, a_post[i,])
}
```

The posterior mean is:

```
apply(lambda_rep, c(2,3), mean)
```

```
          STW       UTW          Ex          TF
STW 0.7655220 0.2201574 0.007058241 0.007262319
UTW 0.1190883 0.8555557 0.006661674 0.018694280
Ex  0.3324237 0.3249717 0.009892477 0.332712192
TF  0.0000000 0.0000000 0.000000000 1.000000000
```

Note that the posterior for the transition probabilities out of Ex reflects the strong prior we specified, given the lack of data. The posterior distributions for the transition probabilities out of the other states are dominated by the information from the observed data.
Note also that the degree of precision in the estimate of the posterior mean depends on the number of Monte Carlo samples drawn (here 1000). In practical applications, you should generate enough samples to obtain the results to the degree of precision that you want.

c. Given these transition probabilities, generate a sample from the posterior distribution of the probabilities of occupying each state for each of cycles 1 to 12 of a Markov model, assuming everyone starts in state STW.

```
J <- 12
nsim <- 1000
S <- 4
pi <- array(dim = c(nsim, S , J+1))
for (i in 1:nsim){
    pi[i,,1] <- c(1,0,0,0)
    for (j in 2:(J+1)){
        for (s in 1:S){
            pi[i,s,j] <- pi[i,,j-1] %*% lambda_rep[i,,s]
        }
    }
}
```

Note the "inner product" operator, `%*%` here used as an alternative way of writing

```
sum(pi[i,,j-1] * lambda_rep[i,,s])
```

We could also write this more cleanly by replacing the product of a vector $\times$ a vector with the product of a vector $\times$ a matrix. It should also be possible to replace the loop over posterior samples `i` with vectorised code — this will make it faster in big models, but probably less clear. See also `heemod` (`https://cran.r-project.org/web/packages/heemod/index.html`) for a packaged method for Markov and similar models.

```
for (i in 1:nsim){
    pi[i,,1] <- c(1,0,0,0)
    for (j in 2:(J+1)){
        pi[i,,j] <- pi[i,,j-1] %*% lambda_rep[i,,]
```

```
    }
}
```

The posterior mean at each time can be computed as follows. Note the increasing proportion of people in TF as time goes on, and a convergence in the relative proportion of people in STW and UTW

```
apply(pi, c(2,3), mean)
```

```
     [,1]        [,2]        [,3]        [,4]        [,5]        [,6]
[1,]    1 0.765522020 0.615253855 0.516896158 0.451724367 0.407824477
[2,]    0 0.220157421 0.358574004 0.443652361 0.494717104 0.524074630
[3,]    0 0.007058241 0.006927038 0.006784531 0.006652072 0.006527573
[4,]    0 0.007262319 0.019245103 0.032666950 0.046906457 0.061573320
            [,7]        [,8]       [,9]       [,10]       [,11]
[1,] 0.377597444 0.356186830 0.34048289 0.328490423 0.318926761
[2,] 0.539574048 0.546230657 0.54725471 0.544704518 0.539901881
[3,] 0.006409081 0.006295237 0.00618511 0.006078065 0.005973665
[4,] 0.076419428 0.091287277 0.10607729 0.120726994 0.135197693
           [,12]       [,13]
[1,] 0.310965602 0.304073491
[2,] 0.533696810 0.526636567
[3,] 0.005871607 0.005771681
[4,] 0.149465981 0.163518262
```

d. Extend the code to draw from the posterior distribution of the (undiscounted) expected costs and health effects over 12 weeks, assuming costs of 7.96, 7.96, and 1000 for STW, UTW and Ex respectively, and a utility of 1 in STW and 0 otherwise.
Solution: as in the lecture. We save only a vector of `nsim` *simulations from the total costs and health effects summed over all times and states.*

```
state_costs <- c(STW=7.96, UTW=7.96, Ex=1000, TF=NA)
state_util <- c(STW=1, UTW=0, Ex=0, TF=0)
cost <- eff <- numeric(nsim)
for (i in 1:nsim){
    cts <- ets <- numeric(J)
    for (j in 1:J){
        state_costs["TF"] <- state_costs[1:3] %*% pi[i,1:3,j]
        cts[j] <- sum(pi[i,,j] * state_costs) # costs for time J, summed over states
       ets[j] <- sum(pi[i,,j] * state_util)  # effects for time J, summed over states
    }
    cost[i] <- sum(cts) # costs summed over all times and states, for this PSA sample
    eff[i] <- sum(ets)  # effects ...
}
summary(cost)
```

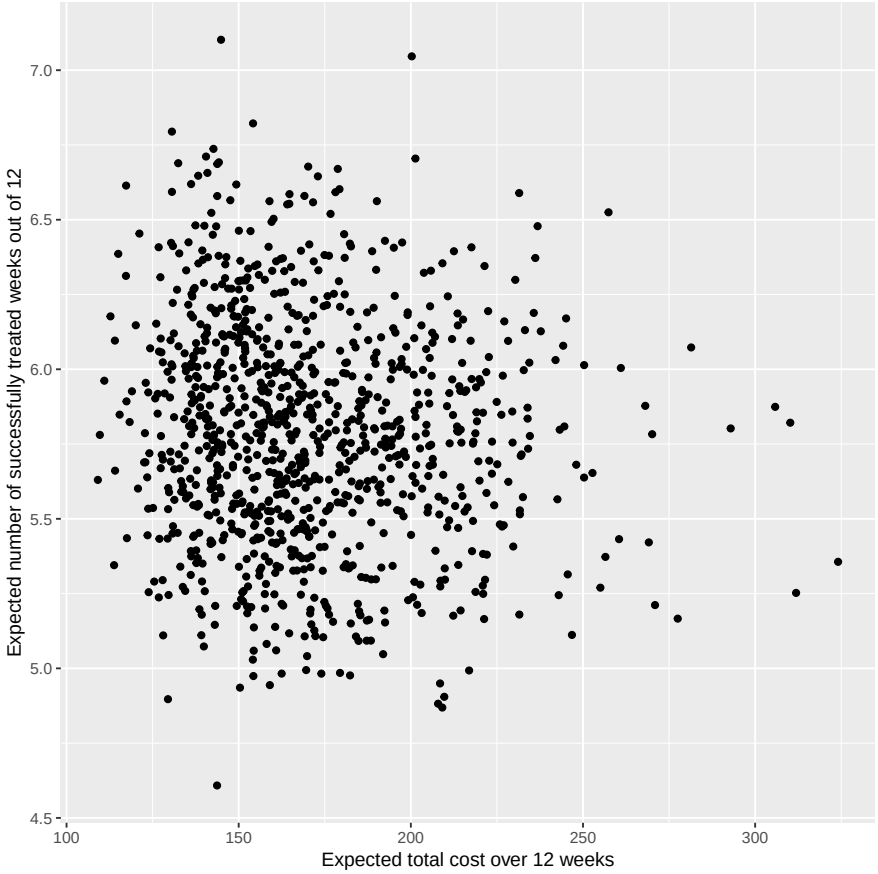
```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  109.1   145.8   164.1   170.3   189.3   324.1
```

```
summary(eff)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  4.608   5.527   5.785   5.790   6.039   7.102
```

e. Plot the joint distribution of costs and effects for the SFC group as a scatterplot in the cost-effectiveness plane.

```
library(ggplot2)
qplot(cost, eff,
      xlab="Expected total cost over 12 weeks",
      ylab="Expected number of successfully treated weeks out of 12")
```




```
## alternatively using base R graphics, simply:
## plot(cost, eff)
```

f. If you have time, repeat the above analysis for the FP group. Thus compute the incremental costs and effects, and compare with the cost-effectiveness scatterplot given in the lecture. Use costs of 2.38 for STW, UTW, and 1000 for Ex, and the same utilities as before
Solution: *First read in the transition count data, compute the posterior parameters, using the same prior, and draw a sample from the posterior of the transition probabilities*

```
N <- rbind(
    "STW"=c(66, 32, 0, 2),
    "UTW"=c(42, 752, 5, 20),
    "Ex"=c(0, 4, 1, 0),
    "TF"=c(0,0,0,0)
)
# (this creates a matrix N with all the counts)
```

```
N
```

```
    [,1] [,2] [,3] [,4]
STW   66   32    0    2
UTW   42  752    5   20
Ex     0    4    1    0
TF     0    0    0    0
```

```
a_post <- a + N
lambda_rep_fp <- array(dim=c(1000, 4, 4))
dimnames(lambda_rep_fp) <- list(NULL, state_names, state_names)
for (i in 1:4){
    lambda_rep_fp[,i,] <- rdiric(1000, a_post[i,])
}
```

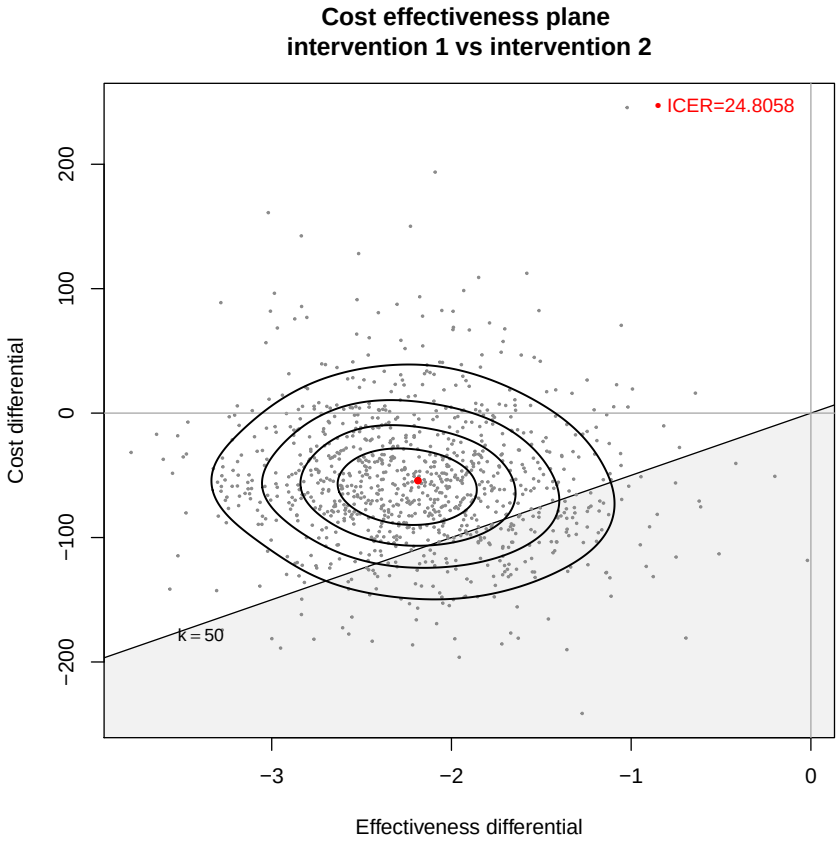
Then compute the state prevalences at each time, and finally compute the total expected costs and effects

```
pi_fp <- array(dim = c(nsim, S , J+1))
for (i in 1:nsim){
    pi_fp[i,,1] <- c(1,0,0,0)
    for (j in 2:(J+1)){
        for (s in 1:S){
            pi_fp[i,s,j] <- pi_fp[i,,j-1] %*% lambda_rep_fp[i,,s]
        }
    }
}
state_costs_fp <- c(STW=2.38, UTW=2.38, Ex=1000, TF=NA)
cost_fp <- eff_fp <- numeric(nsim)
for (i in 1:nsim){
    cts <- ets <- numeric(J)
    for (j in 1:J){
        state_costs_fp["TF"] <- state_costs_fp[1:3] %*% pi_fp[i,1:3,j]
        # costs for time J, summed over states
        cts[j] <- sum(pi_fp[i,,j] * state_costs_fp)
        # effects for time J, summed over states
        ets[j] <- sum(pi_fp[i,,j] * state_util)
    }
    # costs & effects summed over all times and states, for this PSA sample
    cost_fp[i] <- sum(cts)  # costs
    eff_fp[i] <- sum(ets)   # effects
}
```

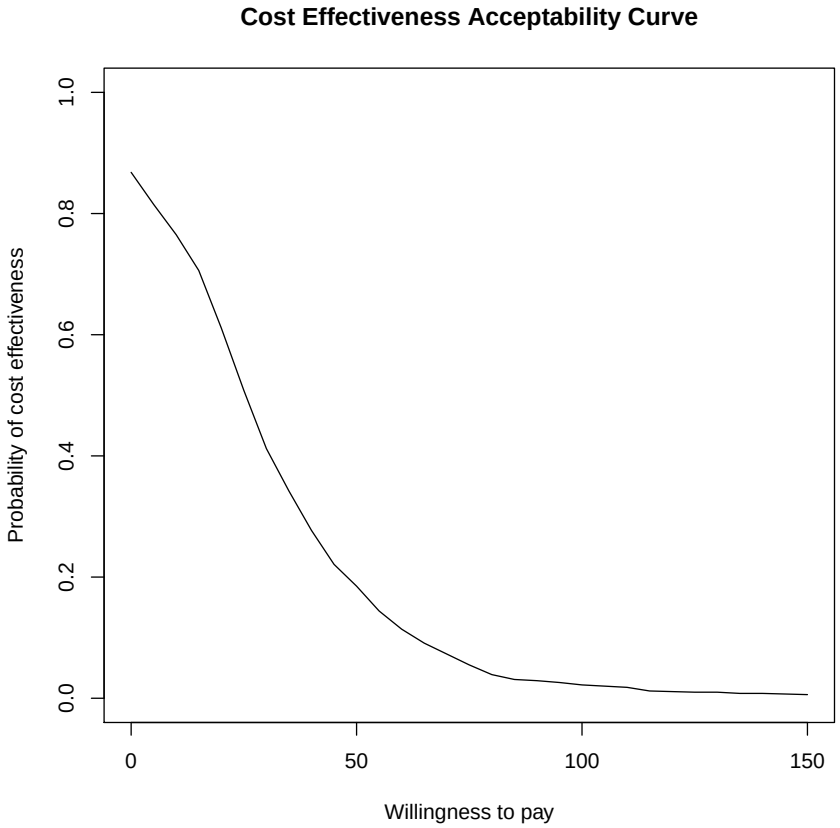
Finally we draw a scatterplot of incremental costs against incremental effects, and a CEAC at willingness-to-pay values from 0 to 150, using the BCEA package. Note in this code, FP is defined as the comparator, while the graph in the lectures was obtained from the 5-state model with SFC as the comparator. The incremental costs and effects are similar to the 5-state model. However we would probably prefer the 5-state version in practice, since we know that hospital and primary care costs are vastly different.

```
library(BCEA)
bc <- bcea(e = cbind("FP"=eff_fp, "SFC"=eff),
           c = cbind("FP"=cost_fp, "SFC"=cost),
```

```
          wtp = seq(0, 150, 5))
contour2(bc, wtp=50)
```




```
ceac.plot(bc)
```



Further reading: Thom et al. PharmacoEconomics. 2017;35(9):951–62, discuss statistical methods to formally assess whether it is worth merging states of a Markov model on the basis of whether the associated transition probabilities, costs and/or utilities are similar.