Practical: Multi-State Markov Models

SPE 2006, Tartu

1 Introduction

The msm package by Christopher Jackson allows you to fit multi-state Markov models. In addition to the online help, there is an introductory manual for msm in PDF format, which is installed with the package. This practical uses examples taken from the manual, but only shows a selection of the capabilities of the msm package. If you want further information after you have finished the practical, search for the file `msm-manual.pdf` using the Windows search facility. You can read it using Adobe Acrobat Reader.

2 The heart data set

The data frame `heart` is provided with the msm package. It describes the follow up history of 622 patients following a heart transplant, monitoring the development of a post-transplant complication called *coronary allograph vasculopathy* (CAV). You can make it available by typing `data(heart, package="msm")` and see a description of the contents with `help(heart, package="msm")`. The reason we are using the `package` argument here is that there is another, completely different data set called "heart" in the survival package. If both msm and survival packages are loaded, and you do not use the package argument, then you will get whichiver one is first on your search path.

We are concerned with modelling the variable `state` which takes values from 1 to 4:

1. healthy
2. mild CAV
3. severe CAV
4. death

and how this evolves with time since transplant (variable `years`).

Before fitting any models, you should spend some time exploring the data in a more informal way. Here is a list of possible questions:

- How old were the subjects when they had their transplant?
- How many were male and how many were female?
- How many visits did the subjects have?
- How long were they followed up for?
- What is the interval between visits?

The `statetable.msm` function tabulates the transitions that are observed between consecutive clinical visits.

```
statetable.msm(state, subject = PTNUM, data = heart)
```

How many deaths occurred in total?

If we were only interested in mortality, and not CAV, we could treat this as a survival analysis problem. Create a new data frame containing only the last visit of each subject. Plot a Kaplan-Meier curve of post-transplant survival.

This analysis is incomplete, since CAV may be an important predictor of mortality. An improved analysis is given below.

3 Fitting a multi-state model

The first task in fitting a multi-state model is to define which transitions are permitted. This can be done by creating a square matrix taking value 1 for the permitted transitions and 0 for the forbidden ones.

```
qmat0 <- matrix(c(0,1,0,1,
                  1,0,1,1,
                  0,1,0,1,
                  0,0,0,0),
                nrow = 4, ncol = 4, byrow=TRUE,
                dimnames=list(from=1:4,to=1:4))
```

Print this matrix. Draw a graph, by hand, with four nodes, representing the states, and arrows between nodes representing the allowed transitions.

A crude estimate of the transition rates can be made with

```
crudeinits.msm(state ~ years, subject=PTNUM, data=heart, qmatrix=qmat0)
```

This function ignores the interval-censoring in the data by assuming that transitions occur at the time of clinical visits. As the function name suggests, it is designed to provide initial values for the model. Save the result as `qmat1`.

The model itself is fitted with a similar function call.

```
heart.msm <- msm(state ~ years, subject=PTNUM, data=heart,
                 qmatrix=qmat1, death=4)
```

By giving the argument `death=4`, we specify that state 4 is a special state, whose transition times are not interval censored: unlike the other states, the time of transition to state 4 (i.e. time of death) is known exactly.

The argument `qmatrix` has a dual purpose: it specifies which transitions are allowed **and** gives initial values for the transition intensities.

Print the `heart.msm` object and use the `summary` function. The output may not be especially useful, so some *extractor* functions are provided to abstract useful statistics from the output:

- `sojourn.msm(heart.msm)` gives the mean *sojourn time* for each state. This means the amount of time spent in each disease state before moving on to the next one.
- `plot(heart.msm)` Plots parametric survival curves, stratified by disease state. These show how more severe CAV is associated with higher mortality.
- `pmatrix.msm(heart.msm, t)` creates the transition probability matrix for a time interval t. If P is the transition matrix then $P_{ij}(t)$ gives the probability of a subject being in state j at time t, given that they were in state i at time 0. Use this to calculate the proportion of transplant patients that we expect to be healthy 1, 5, and 10 years after transplant.

4 Adding covariates

We shall consider whether transition rates are different for males and females. Before doing so, save the estimated transition intensity matrix from the previous model, to act as a new set of initial values

```
qmat2 <- qmatrix.msm(heart.msm)$estimate
```

To examine the effect of sex, supply a formula to the `covariate` argument

```
heart.msm.sex <- msm(state ~ years, subject=PTNUM, data=heart,
                     qmatrix=qmat2, death=4, covariate = ~ sex)
```

Then

```
hazard.msm(heart.msm.sex)
```

gives hazard rate ratios, and confidence intervals, for the allowed transitions.

5 Applying constraints

By default, the effect of the covariates is assumed to be different for each possible transition. In this case, there are 7 possible transitions, and therefore 7 hazard rate ratio estimates. We can also simplify the model, to answer some more clinically relevant questions:

1. Is the mortality rate higher (or lower) for females?
2. Is the rate of CAV progression higher?
3. Is the rate of CAV recovery higher?

This is done by supplying a `constraint` argument

```
constraint = list(sex = c(1,2,3,1,2,3,2))
```

The `constraint` argument gives a numeric label to the allowed transitions (reading across the rows of the qmatrix), and constrains the effect of the covariate to be the same for all transitions with the same label. The labels are shown as super-scripts below:

$$\begin{pmatrix} 0 & 1^1 & 0 & 1^2 \\ 1^3 & 0 & 1^1 & 1^2 \\ 0 & 1^3 & 0 & 1^2 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Fit this constrained model to the `heart` data and answer the above questions.

6 A misclassification model

So far, we have assumed that any observed improvements in CAV state are real (*i.e.* transitions from state 2 to 1 and from state 3 to 2 really do take place). An alternative interpretation is that CAV is a *progressive* disease, which can only get worse with time, and that any apparent improvements in disease status are due to misclassification. The msm package can also fit such models.

Consider a model in which improvements in disease status are no longer possible. Draw a new graph for this model, and write down a new matrix, equivalent to `qmat0` with 1 for allowed transitions and 0 for the forbidden transitions. Create a new `qmatrix` of initial values for the new model, using the estimated transition intensities of the previous model.

You also need to create a misclassification matrix `ematrix` with the following values

$$\begin{pmatrix} 0 & 0.1 & 0 & 0 \\ 0.1 & 0 & 0.1 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

The `ematrix` has a similar interpretation as `qmatrix`, but applies to misclassification probabilities instead of transition intensities. Zero values indicate impossible misclassification. The fourth row consists entirely of zeros, showing that death cannot be misclassified. Likewise, the fourth column is zero, indicating that states 1-3 cannot be misclassified as death. Non-zero values indicate possible misclassifications, and also supply initial values for the model. Hence, for example: if the true disease state is 1 (healthy), we may observe state 2 (mild CAV) instead. Initially, the misclassification probability is assumed to be 10%, but the model will calculate a new estimate during the course of model fitting.

Fit a new model, without covariates, using the new qmatrix and with an extra argument

```
ematrix = ematrix
```

Use the `ematrix.msm` function to extract the estimated misclassification matrix. Plot the new survival curves and compare them with the previous ones.

What is the influence of sex on the progressive model?