Matrix exponentials and survival curves

Terry Therneau

May 2023

1 Simple Cox models

Define

$$\begin{aligned} N(t) &= \sum_i N_i(t) \\ Y(t) &= \sum_i Y_i(t) \\ \lambda(t) &= dN(t)/Y(t) \end{aligned}$$

where $N_i(t)$ is the cumulative number of events up to time t for subject i, and $Y_i(t)$ is the 0/1 indicator that subject i is at risk.

For a survival curve we almost uniformly use the Kaplan-Meier estimate, with the Fleming-Harrington as a rare alternative:

$$\begin{aligned} KM(t) &= \prod_{s \le t} (1 - \lambda(s)) \\ FH(t) &= \prod_{s \le t} e^{-\lambda(s)} \\ &= \exp(-\sum_{s \le t} \lambda(s)) \end{aligned}$$

Since $\exp(-x) \approx 1 - x$ we could view the KM as a first-order Taylor series approximation to the FH.

For survival curves based on a Cox model things get a bit more complicated. For a fit with covariates x and coefficient β, the predicted hazard for a new subject with covariate vector z will be

$$\lambda(t; z) = \frac{dN(t)}{\sum_i Y_i(t) e^{(x_i - z)'\beta}}$$

and the Breslow estimate of survival is

$$S(t; z) = \prod_{s \le t} e^{-\lambda(s;z)}$$

This is parallel to the Fleming-Harrington form. The KM equivalent is never used for a simple reason, which is that for large values of z (assume wlog that β is positive) $\lambda(t;z)$ may be greater than 1, which in turn leads to a negative values of $1-\lambda$, i.e., negative values for $S(t)$. This most often occurs near the end of the curve, when the number at risk is small.

2 Multi-state models

For a multi-state hazards model with m states, there will be a family of hazards, one for each possible transition from state j to state k ($j \neq k$).

$$\lambda_{jk}(t) = \frac{dN_{jk}(t)}{\sum_i Y_{ij}(t)}$$
$$\lambda_{jk}(t;z) = \frac{dN_{jk}(t)}{\sum_i Y_{ij}(t) e^{(x_i - z)'\beta}}$$

N_{jk} counts the cumulative number of j to k transitions, and $Y_{ij}(t)$ is 1 if subject i is in state j and at risk for a transition (out of state j) at time t. The first equation above defines the non-parametric hazard, and the second the predicted hazard from a multi-state hazards model. The coefficicient vector β will often be different for each j:k transition, but we have omitted that from the notation for simplicity.

The m by m intensity matrix $A(t)$ is defined to have off diagonal elements $A_{jk}(t) = \lambda_{jk}(t)$ and similarly for $A(t;z)$ based on the multi-state hazards (Cox) model. The diagonal element is defined such that each row sums to zero, $A_{jj} = -\sum_{k \neq j} \lambda_{jk}(t)$. Two natural estimates of the probability in state are the Aalen-Johansen and exponential estimates

$$AJ(t) = p(0) \prod_{s \leq t} (I + A(s))$$
$$p(t) = p(0) \prod_{s \leq t} e^{A(s)}$$

Here $p(0)$ is the probabilty distribution at the starting time, which is very often $(1, 0, 0, \ldots, 0)$, i.e., everyone starts in state 1, and e is the matrix exponential. For matrices, $\exp(A)\exp(B)! = \exp(A+B)$ unless $AB = BA$, a condition that will not hold for our models, so the second equation does not collapse to a sum. For both estimates $\sum p(t) = 1$ at all time points.

For a multi-state Cox model, the first formula has the same flaw as before, namely that for some values of z it will lead to negative elements in $p(t)$. This normally occurs for high risk subjects (high predicted hazards for one or more transitions) and smaller risk sets; the flaw normally does not arise in a study with significant censoring as the risk sets never become small. However, unlike the single endpoint Cox model, this alternate estimate *is* suggested as an estimate by many authors, e.g. the Cook and Lawless textbook, and also appear as the default in some packages (mstate). The survival package follows the Breslow pattern and uses the exponential estimate.

Accurate and efficient compuation of the matrix exponential is a long-standing research topic. The simple and direct definition

$$e^A = I + A + A^2/2! + A^3/3! + A^4/4! + \ldots$$

is neither efficient or accurate. A strong background is provided by the textbook of Higham [?]; many of the methods discussed there and some newer refinements are incorporated into the expm library in R. One important special case is for an event time where all the transitions are from a single state. The intensity matrix A then will have only a single non-zero row, say it is row j. Then $B = \exp(A)$ will be equal to the identity matrix, for all rows except row j, and

$$B_{jj} = exp(A_{jj})$$
$$B_{jk} = (1 - B_{jj})\lambda_{jk} / \sum_{k \neq j} \lambda_{jk}$$

This occurs for any event time without ties, and also for competing risks. It is so common that the survival package uses an internal routine `survexpm` which checks for the case, invoking the expm routine otherwise.

One interesting aside is the issue of scaling. The identity $\exp(\theta I + A) = exp(\theta)\exp(A)$ means that it is easy to pre-scale the diagonal of A, which can have an impact on downstream compuations. Corollary 4.22 of [?] shows that for an intensity matrix A, the optimal scaling is $\theta = \max_j |A_{jj}|$; the resulting matrix $B = \theta I + A$ has all elements positive. Hence B^k is postive for all powers k and there is no cancellation in the simple Taylor series for expm.

An $A(t; z)$ matrix that actually occurs in one of our examples has

```
> A = rbind(c(-.2, .1, .1), c(0, -1.1, 1.1), c(0, 0,0))
> expm(A)
3 x 3 Matrix of class "dgeMatrix"
          [,1]       [,2]      [,3]
[1,] 0.8187308 0.05398441 0.1272848
[2,] 0.0000000 0.33287108 0.6671289
[3,] 0.0000000 0.00000000 1.0000000
> B <- A + 1.1*diag(3)
> exp(-1.1) * expm(B)   # verify the formula
3 x 3 Matrix of class "dgeMatrix"
          [,1]       [,2]      [,3]
[1,] 0.8187308 0.05398441 0.1272848
[2,] 0.0000000 0.33287108 0.6671289
[3,] 0.0000000 0.00000000 1.0000000
> diag(3) + A   # the bad estimate
     [,1] [,2] [,3]
[1,]  0.8  0.1  0.1
[2,]  0.0 -0.1  1.1
[3,]  0.0  0.0  1.0
> diag(3) + A + A^2/2 + A^3/6
          [,1]      [,2]      [,3]
[1,] 0.8186667 0.1051667 0.1051667
[2,] 0.0000000 0.2831667 1.9268333
[3,] 0.0000000 0.0000000 1.0000000
> exp(-1.1) *(diag(3)+ B)
```

```
          [,1]       [,2]       [,3]
[1,] 0.6324551 0.03328711 0.03328711
[2,] 0.0000000 0.33287108 0.36615819
[3,] 0.0000000 0.00000000 0.69902928
> exp(-1.1) *(diag(3)+ B + B^2/2 + B^3/6)
          [,1]       [,2]       [,3]
[1,] 0.8077117 0.03500694 0.03500694
[2,] 0.0000000 0.33287108 0.64138710
[3,] 0.0000000 0.00000000 0.97425818
```

We see that the scaled version converges faster, but $I + B$ is still not a particularly good approximation.