Penalisation Methods for Multi-State Models – A Comparative, Simulation Study

Ms. Chantelle Cornett (CC), Dr. Glen Martin (GM), Dr. Niels Peek (NP)

Abstract

1. Introduction

Within medical studies patients are often observed over time, and their information captured on multiple occasions. When individuals can experience several events, multi-state models are usually used to analyse this type of data. Multi-state models (MSMs) are useful for prediction of complex clinical settings where patients may transition through several intermediate states before experiencing a terminal event [1-5]. A challenge of MSMs is the combinatorial complexity in models as the number of states being modelled grows. This increases computation time, but also risks overfitting of some of the transition models within the MSM, especially those with fewer events/ observations.

Overfitting refers to when a model is too complex and captures noise/random fluctuations in the data rather than the true relationships between variables. In the case of MSMs, this usually manifests as too many parameters and overly complex model structures. One way overfitting can be diagnosed for MSMs is to assess the model's performance on data that was not used in the training process. Techniques such as cross-validation and comparison of prediction accuracy metrics such as the C-index between the training and test sets can aid in diagnosing overfitting in an MSM [6]. Overfitting in this context has been recognised as an issue in multi-state models [7][8].

One potential solution to this is to share parameters between transitions within the model [5]. For example, one could assume that certain groups of transitions share a common baseline and/or predictor effects. The properties of doing this in a prediction context are unknown. Likewise, in a prediction context, penalisation methods are often advocated to help minimise the risk of overfitting. Therefore, it is of interest to examine the impact of penalisation methods on this situation with or without sharing of parameter terms across the transitions. While the use of shared parameters across transitions to reduce the parameter space is well-documented, the concept of penalisation in this context is less well-studied.

Although sample size requirements for MSMs have not yet been developed, it is known that sample size is an important factor in determining a well-functioning model [9][10][11]. While penalisation may prove to be a useful tool to avoid overfitting, many papers have noted that penalisation is not a substitute for an adequate sample size [12][13][14]. Therefore, we will not go below the minimum sample size for a Cox model developed for each transition (in the absence of the MSM sample size criteria). We will explore the use of penalisation and parameter sharing as further protection against overfitting.

This paper is organised as follows. Section 2 will explain the methodological concepts surrounding MSMs, with emphasis on the estimation of the covariate effects and transition probabilities. In this section we will also discuss some recent methods proposed for the penalisation of MSMs. Section 3 will discuss the data generating mechanism used for simulating the data and the packages used in R for applying each of the penalisation methods. The results from applying each of the penalisation methods will then be compared in Section 4, before being discussed in Section 5. Concluding remarks will then be made in Section 6.

2. Multi-State Models

2.1 Definition

A multi-state process is defined as a stochastic process $(X(t), t \in T)$, where $T = [0, \tau], \tau < \infty$ is a time interval and the value of the process at time t is the state (contained in a finite state space $S = \{1, \dots, N\}$) occupied by the process at that time. As the multi-state process progresses, a history H_{t^-} will be generated. This history will consist of the states the process visits over the interval $[0, t\}$ and the time of the transitions. The process can be characterised through transition probabilities between states h and j,

$$P_{hj}(s,t) = \mathbb{P}(X(t) = j \,|X(s) = h, H_{s^-}),$$

for $h, j \in S,\ \ s, t\ \in T, s \leq t$. It can also be characterised through transition intensities,

$$\alpha_{hi}(t) = \lim_{\Delta t \to 0} \frac{P_{hi}(t, t+\Delta t)}{\Delta t},$$

which represent the instantaneous hazard of the progression to state j from state h. Both characterisations depend on the history of the process.

Though there are different assumptions that can be made about the dependence of transition rates over time, in this paper we will assume a time homogeneous model, meaning the intensities are constant over time.

2.2 Recently Proposed Penalisation Methods

Commented [CC1]: Need to complete

3. Method

All analysis for this study will be conducted using R (version 4.3.1) in the RStudio environment under the MacOS Ventura operating system on a MacBook Pro (with Apple M1 Pro processor). Code for simulations and models will be provided on GitHub to ensure reproducibility. Before applying any of the penalisation methods, we need to simulate our data.

3.1 Simulation

We will consider two data generating models: first, a simple illness-death model, and second, a more complex 5-state model. This is done so that we can investigate the impact of the number of states (and transitions) on the predictive performance results. Each simulation will have n=200,000 individuals. In order to compromise between reducing the Monte-Carlo standard errors and having a reasonable computation time, we will repeat the simulation 500 times ($n_{sim} = 500$).

Data will need to be passed in a long data format. We will first simulate the model in wide format for ease of computation, and transpose this to create the long format we require.

3.1.1 Simple Illness-Death Model

We will start by simulating data for a simple illness-death model (Figure 1). We will use the following Q-matrix:

$$Q = \begin{bmatrix} -0.02 & 0.015 & 0.005 \\ 0 & -0.01 & 0.01 \\ 0 & 0 & 0 \end{bmatrix},$$

using the *simmulti.msm* function from the *MSM* package to get a list of states and their transition times (starting in the healthy state at time 0) for each subject. The numbers in this Q-matrix were chosen in order to get sufficient progress through the states in the model while compromising on computation time.
The survival distributions for each transition were drawn from a hazard function of:

$$\exp[0.5 * age_i - 0.5 * gender_i + 0.5 * BMI_i] * \lambda_{j,k}$$

The baseline hazard for each transition ($\lambda_{j,k}$) was assumed to follow a Weibull distribution, where parameters were chosen based on the mean event time:
$\lambda_{1,2} = t^{-0.791}$ (λ =1, k = 0.219)
$\lambda_{1,3} = t^{-0.791}$ (λ =1, k = 0.219)
$\lambda_{2,3} = t^{-0.766}$ (λ =1, k = 0.233)

Commented [CC2]: changed

We will base the structure of the data on the structure of the European Society for Blood and Marrow Transplantation dataset (ebmt4)[18]. The final dataset will have the following columns before being passed into *msprep* to transform the dataset into a format suitable for use by the *mstate* package:

Column	Description
Subject	The subject IDs.
timeToIll	The time passed since starting in the healthy state until being in the ill state.
timeToDeath	The time passed since starting in the healthy state until being in the death state (if death state is not reached this will take the value of 200, indicating censoring).
illStat	An indicator variable taking a value of 1 if the subject reaches the ill state, and 0 if they do not.
deathStat	An indicator variable taking a value of 1 if the subject reaches the death state, and 0 if they do not.
Age	A factor variable, taking values ≤20, 20-40, ≥40. These are assigned to subjects with probabilities 0.332, 0.531, 0.137 in accordance with the estimated worldwide age distribution [19].
Gender	A factor variable, taking values 0 and 1 and assigned with equal probability.

BMI	A continuous variable, indicating the body mass index of an individual. This will be drawn from a standard Gaussian distribution.

Table 2: A list of columns for the simple illness-death model and their descriptions.

3.1.2 5-State Model

The next model we will simulate is a 5-state model (Figure 2). We will use the following Q-matrix:

$$Q = \begin{bmatrix} -0.025 & 0.02 & 0 & 0 & 0.005 \\ 0 & -0.035 & 0.025 & 0 & 0.01 \\ 0 & 0 & -0.035 & 0.025 & 0.01 \\ 0 & 0 & 0 & -0.035 & 0.035 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

using the *simmulti.msm* function from the *MSM* package to get a list of states and their transition times (starting in the healthy state at time 0) for each subject. The numbers in this Q-matrix were chosen in order to get sufficient progress through the states in the model while compromising on computation time.
The survival distributions for each transition were drawn from a hazard function of:

$$\exp[0.5 * age_i - 0.5 * gender_i + 0.5 * BMI_i] * \lambda_{j,k}$$

The baseline hazard for each transition ($\lambda_{j,k}$) was assumed to follow a Weibull distribution, where parameters were chosen based on the mean event time:
$\lambda_{1,2} = t^{-0.705}$ (λ =1, k = 0.295)
$\lambda_{2,3} = t^{-0.690}$ (λ =1, k = 0.310)
$\lambda_{3,4} = t^{-0.666}$ (λ =1, k = 0.333)
$\lambda_{4,5} = t^{-0.625}$ (λ =1, k = 0.375)
$\lambda_{1,5} = t^{-0.752}$ (λ =1, k = 0.248)
$\lambda_{2,5} = t^{-0.742}$ (λ =1, k = 0.258)
$\lambda_{3,5} = t^{-0.729}$ (λ =1, k = 0.271)

Commented [CC3]: changed

We will base the structure of the data on the structure of the ebmt4 dataset [18]. The final dataset will have the following columns, before being passed into *msprep* to transform the dataset into a format suitable for use by the *mstate* package:

Column	Description
Subject	The subject IDs.
timeToMild	The time passed since starting in the healthy state until being in the mild state.
timeToModerate	The time passed since starting in the healthy state until being in the moderate state (if moderate state is not reached this will take the value of 200, indicating censoring).
timeToSevere	The time passed since starting in the healthy state until being in the severe state (if severe state is not reached this will take the value of 1000, indicating censoring).

timeToDeath	The time passed since starting in the healthy state until being in the death state (if death state is not reached this will take the value of 1000, indicating censoring).
mildStat	An indicator variable taking a value of 1 if the subject reaches the mild state, and 0 if they do not.
modStat	An indicator variable taking a value of 1 if the subject reaches the moderate state, and 0 if they do not.
sevStat	An indicator variable taking a value of 1 if the subject reaches the severe state, and 0 if they do not.
deathStat	An indicator variable taking a value of 1 if the subject reaches the death state, and 0 if they do not.
Age	A factor variable, taking values ≤20, 20-40, ≥40. These are assigned to subjects with probabilities 0.332, 0.531, 0.137 in accordance with the estimated worldwide age distribution [19].
Gender	A factor variable, taking values 0 and 1 and assigned with equal probability.
BMI	A continuous variable, indicating the body mass index of an individual. This will be drawn from a standard Gaussian distribution.

Table 3: A list of columns for the 5-state model and their descriptions.

3.2 Penalisation Methods

3.2.1 MLE No Shrinkage

In order to apply the MLE without shrinkage, the *msprep* function of the *mstate* package will be used to prepare the data for the *coxph* function. This facilitates fitting a cox proportional hazards model for each transition in turn.

3.2.2 MLE with Uniform Shrinkage Factor Applied to Each Transition

To apply a uniform shrinkage factor to each transition, we will estimate the shrinkage factor post model fit using the heuristic shrinkage factor:

$$S_{VH} = 1 - \frac{p}{LR},$$

where p is the total number of predictor parameters for the full set of candidate predictors and LR is the likelihood ratio (chi-squared) statistic for the fitted model defined as:

$$LR = -2(lnL_{null} - lnL_{model}),$$

where lnL_{null} is the log-likelihood of a model with no predictors (e.g. intercept-only cox proportional hazards model), and lnL_{model} is the log-likelihood of the final model [20]. In the model, we will multiply each covariate by the shrinkage factor, allowing the shrinkage factor to be applied uniformly to both covariates across each transition in the model. This will be fitted using the *mscoxph* function from the *mstate* package.

3.2.3 LASSO Penalised Likelihood

The least absolute shrinkage and selection operator (LASSO, also known as ℓ_1 regularisation) aims to minimise:

$$F(\beta) = \sum_{i=1}^{n}(y_i - x_i{}^T\beta)^2 + \lambda\sum_{j=1}^{p}|\beta_j|,$$

where $\lambda \geq 0$ is a tuning parameter, that can be chosen using cross-validation.

In order to apply the LASSO penalised likelihood, we will first concentrate on finding the optimum tuning parameter for the LASSO. This will be done using cross-validation:

1. Split the dataset into 10 folds.
2. For each potential value of the tuning parameter, perform a cross-validation loop.
3. Choose the tuning parameter that provides the best model performance across all folds.

In R, this will be done using the *penMSM* package, using the *penMSM function* with *type = lasso* to perform the cross-validated LASSO.

3.2.4 Fused-LASSO Penalised Likelihood (Shared Parameters)

The fused least absolute shrinkage and selection operator (fused-LASSO) aims to minimise:

$$F(\beta) = \frac{1}{N}\sum_{i=1}^{N}(y_i - x_i{}^T\beta)^2 ,$$

subject to $\sum_{j=1}^{p}|\beta_j| \leq t_1$ and $\sum_{j=2}^{p}|\beta_j - \beta_{j-1}| \leq t_2$.

The first constraint is the same constraint as in the LASSO penalised likelihood, while the second constraint penalises large changes to temporal structure, forcing any coefficients to vary smoothly.

Just like in the LASSO penalised likelihood, we will first find the optimum tuning parameter for the fused-LASSO penalisation. When using the *penMSM* function, we will set the *type* argument to *fused* to indicate a fused-LASSO approach.

3.2.5 Reduced-Rank Method

The reduced rank method is a form of pre-penalisation, which encourages some of the parameters to be constrained to be close or equal to zero. It is often applied when there is prior knowledge or belief that the true underlying structure involves a lower-dimension subspace.

A model with time-varying effects of covariates can be described using a structure matrix $\ominus$ which contains the coefficients for the covariates and their interactions with the time functions. This structure matrix can be factorised in different ways, resulting in a matrix with a lower rank r. If we consider B a $p * r$ matrix and Γ a $q * r$ matrix of coefficients that factorise the $\ominus$ matrix as $\ominus = B\,\Gamma'$, then the rank r model can be written as:

$$h(t|X) = h_0(t)\exp\,[XB\Gamma'F'(t)]$$

To implement the reduced-rank method of penalisation, we will use the *redrank* function of the *mstate* package. We will choose the value of *R* (the rank of the matrix used for the reduced-rank problem) based on cross-validation and choose the rank that minimizes prediction error on unseen data.

3.2.6 Bayesian Approach with Penalising Prior (Shared Parameters)

We have decided to use a horseshoe prior as described in Beesley (2020). A horseshoe prior is a continuous shrinkage prior, that strongly shrinks moderate and weak signals to zero, while still allowing very large signals. The prior takes the form:

$$f(\theta_k|\lambda_k,) \; = \; N(\alpha, \lambda_k^2\tau^2)$$
$$f(\lambda_k) = Cauchy^+(0,1)$$
$$f(\tau) = Cauchy^+(0,1),$$

where smaller τ implements a greater global shrinkage, θ_k are the log-hazard ratios for each of the k transition models, α is a non-zero constant (assuming that the 'shared' coefficient is non-zero across all transitions), and $Cauchy^+$ indicated the half-Cauchy distribution. τ, λ_k, and θ_k can be sampled in turn using the Metropolis-Hasting method. As we are applying the horseshoe prior to an MSM, we will define τ separately for each transition.

We will be using the *brm* function from the *brms* package to implement this type of penalisation. We will specify priors for the intercept, standard deviation, and each covariate in the model. The models using the simulated data will not be complex, so we will only use 2 chains. We will start with 2000 (with a burn-in period of 1000 iterations) iterations, as this provides a good balance between computational efficiency. Using trace-plots and the Gelman-Rubin statistic, we will assess convergence; if these diagnostic tools show that insufficient convergence has occurred, we will increase the number of iterations.

3.2.7 Packages to Use

Package	Use
mstate	Contains functions needed for data preparations, description, and hazard estimation for multi-state models [21]. This includes reduced rank proportional hazards model for multi-state models.
tidyverse	A collection of packages for preparing, wrangling, and visualising data [22].
genSurv	Generation of survival data with one (binary) time-dependent covariate [23].
shrink	Used for uniform (global) shrinkage [24].
brms	Used for fitting a Bayesian Cox model with a horseshoe prior [25].
penMSM	Used for providing efficient LASSO penalisation [7].

Table 4: A list of packages and their uses.

4. Results

5. Discussion

6. Conclusion