Plan for Comparative Methods Study on Penalisation Methods for Multi-State Models

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

Overview

Multi-state models (MSM) are useful for prediction of complex clinical settings where patients may transition through several intermediate states before experiencing a terminal event [1][2][3][4][5]. A challenge of MSMs is the combinatorial complexity in models as the number of chronic conditions 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 (of multi-morbidity) 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.

Definitions

We broadly define the terms relevant to this review:
MLE – Maximum likelihood estimate
LASSO - least absolute shrinkage and selection operator
CPRD – Clinical practise research datalink
Fold – A fold refers to the subset of the dataset that is used for training and validation.
Cross validation loop – A cross validation loop is the process by which a dataset is split into k folds. The model would then be trained and validated k times and using a different fold as the validation set each time.
Horseshoe prior – A type of Bayesian prior distribution, designed to encourage shrinkage of parameter estimates toward zero while allowing for the possibility of some parameters to remain large.
Metropolis-Hasting method – A Markov Chain Monte Carlo method (MCMC) used for generating a sequence of random samples from a probability distribution.
Burn-in period – The initial phase of the simulation where the Markov chain is allowed to converge to the target distribution.
Bootstrap sampling – A sample made by randomly drawing observations with replacement from the original dataset. This creates new datasets of the same size as the original but with some observations omitted and others repeated.

Methodology

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.

Aims

This study aims to compare current penalisation methodology for multi-state models. We will be considering the following methods of implementing the penalisation within prediction models:

- MLE no shrinkage
- MLE with uniform shrinkage factor applied to each transition
- LASSO penalised likelihood
- Fused-LASSO penalised likelihood [5]
- Reduced-rank methods [9]
- Bayesian approach with penalising prior to pull coefficients down towards an overall model-level value [10]

These methods will be applied to simulated data, and then applied to CPRD data [11].

Data Simulation

We will start by simulating data for a simple illness-death model. From this, we will expand the model to change the number of states (and consequently, the number of transitions) using the *mstate* package (see Table 1). We will do this in a way such that the number of states and transitions will be determined by a parameter that we can vary across our simulation study.

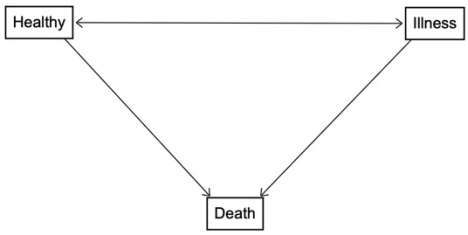

Figure 1 - Simple illness-death model.

Due to restrictions in packages we plan to use, 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.

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:

$$Q = \begin{bmatrix} -0.02 & 0.2 & 0 \\ 0 & -0.01 & 0.01 \\ 0 & 0 & 0 \end{bmatrix},$$

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. We will base the structure of the data on the structure of the European Society for Blood and Marrow Transplantation dataset (ebmt4) [12]. 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:

- 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 [13].
- Gender: A factor variable, taking values 0 and 1 and assigned with equal probability.

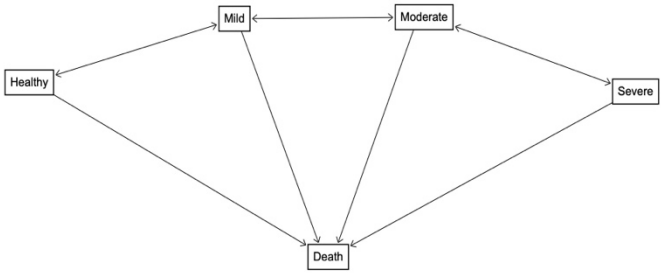

Figure 2 – An example of a 5-state model.

5-State Model

CHECK IF PREVIOUS SECTION IS OK BEFORE WRITING THIS SECTION

Penalisation Methods

Since we are using simulated survival data, a Cox proportional hazards model will be implemented using each of the penalisation methods in turn.

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.

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:

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

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 $\ln L_{null}$ is the log-likelihood of a model with no predictors (e.g. intercept-only logistic regression model), and lnL_{model} is the log-likelihood of the final model [14]. In the model we will multiple 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.

LASSO Penalised Likelihood

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.

Fused-LASSO Penalised Likelihood (Shared Parameters)

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.

Reduced-Rank Method

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.

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 a multi-state model, 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.

Packages to Use

Package	Use
mstate	Contains functions needed for data preparations, description and hazard estimation for multi-state models [15]. This includes reduced rank proportional hazards model for multi-state models.
tidyverse	A collection of packages for preparing, wrangling and visualising data [16].
genSurv	Generation of survival data with one (binary) time-dependent covariate [17].
shrink	Used for uniform (global) shrinkage [18].
brms	Used for fitting a Bayesian Cox model with a horseshoe prior [19].
penMSM	Used for providing efficient LASSO penalisation [7].

Table 1: A list of packages and their uses.

Performance Measures

- Computation time – As combinatorial complexity increases, so does complexity.
- Time-dependent ROC curves – To quantify discrimination.
- C-Index – To evaluate predictive performance.
- Calibration plots – Show any potential mismatches between observed and predicted probabilities in the data.
- Transition-specific calibration plots – To evaluate overfitting.

· Choose a value of nsim that achieves acceptable Monte Carlo SE for key performance measures.

Validation

Internal Validation

For each iteration of the simulation, we will be generating a large ($n \geq 500{,}000$) sample size and will split this into two parts: a development subset and a validation subset. We will ensure that the validation subset is large ($n \geq 200{,}000$). For each penalisation method, we will fit a model using the validation subset using the same model specification as we used in the original analysis.

For each model fitted using the validation subset, we will calculate the following performance measures:

- Time-dependent ROC curves [20]
- C-Index [21]
- Calibration plots [22]
- Transition-specific calibration plots

These will be compared with the performance measures of the original model to check that the results are consistent between the development and validation subsets.

External Validation - Use on 'Real World' Data

Using the CPRD dataset, we will implement the Cox models for each of the penalisation methods using the same methods used to build the original models. We will then calculate the following performance measures:

- Time-dependent ROC curves [20]
- C-Index [21]
- Calibration plots [22]
- Transition specific calibration plots

We will compare the performance of the models using 'real world' data to their equivalent models that used simulated data. If any model's performance using the CPRD dataset [8] is significantly

different from its performance on the original dataset we will consider refitting the model based on the new dataset.

Author Contributions

CC and GM contributed to the conception and design of the study. All authors contributed to discussions around updates to the protocol.

CC wrote the protocol.

CC performed the data simulation and method implementation, iteratively updating the protocol where necessary.

Bibliography

[1] Heinz Freisling et al, 2020. Lifestyle factors and risk of multimorbidity of cancer and cardiometabolic diseases: a multinational cohort study. *National Library of Medicine,* 10 Jan. Harrison Reeder et al, 2022. Penalized Estimation of Frailty-Based Illness-Death Models for Semi-Competing Risks. *Biometrics.*

[2] Hein Putter et al, 2006. Estimation and Prediction in a Multi-State Model for Breast Cancer. *Biometrical Journal,* 16 June, 48(3), pp. 366-380.

[3] Niels Keiding et al, 2001. Multi-state models and outcome prediction in bone marrow transplantation. *Statistics in Medicine,* 30 May, 20(12), pp. 1871-1885.

[4] Jeffrey Cannon et al, 2017. Rheumatic Heart Disease Severity, Progression and Outcomes: A Multi-State Model. *Journal of the American Heart Association,* 2 March.

[5] Holger Sennhenn-Reulen, 2016. Structured fusion lasso penalized multi-state models. *Statistics in Medicine,* 23 June, 35(25), p. 4637-4659.

[6] Jake Lever et al, 2016. Model selection and overfitting. *Nature Methods*, 30 August, 13, p703-704.

[7] Holger Reulen, 2023. Package 'penMSM'. [Online]
Available at: https://cran.r-project.org/web/packages/penMSM/penMSM.pdf
[Accessed 15 11 2023]

[8] Michael Crowther et al, 2017. Parametric multistate survival models: Flexible modelling allowing transition-specific distributions with application to estimating clinically useful measures of effect differences. Statistics in Medicine, 05 September, 36(29), p4719-4742.

[9]Marta Fiocco et al, 2009. Reduced-rank proportional hazards regression and simulation-based prediction for multi-state models. *Statistics in Medicine,* 21 April, 27(21), pp. 4340-4358.

[10] Lauren Beesley et al, 2020. Bayesian variable selection and shrinkage strategies in a complicated modelling setting with missing data: A case study using multistate models. *Statistical Modelling*, 21(1-2), p11-29.

[11] Shivani Padmanabhan, 2011. CPRD GOLD Data Dictionary [Restricted]

[12] Gustavo Soutinho, 2021. ebmt4: Data from the European Society for Blood and Marrow Transplantation. Available at: https://rdrr.io/github/gsoutinho/markovMSM/man/ebmt4.html
[Accessed 19 11 2023]

[13] Carmen Ang, 2021. Visualizing the World's Population by Age Group. Available at: https://www.visualcapitalist.com/the-worlds-population-2020-by-age/
[Accessed 19 11 2023]
[14] Richard Riley et al, 2018. Minimum sample size for developing a multivariable prediction model: PART II - binary. *Statistics in Medicine,* 24 Oct, 38(7), p1276–1296.
[15] Hein Putter et al, 2021. *mstate: Data Preparation, Estimation and Prediction in Multi-State Models.* [Online]
Available at: https://cran.r-project.org/web/packages/mstate/index.html
[Accessed 9 11 2023].
[16] Hadley Wickham, 2023. *tidyverse: Easily Install and Load the 'Tidyverse'.* [Online]
Available at: https://cran.r-project.org/web/packages/tidyverse/index.html
[Accessed 9 11 2023].
[17] Arthur Araujo, 2021. *genSurv: Generating Multi-State Survival Data.* [Online]
Available at: https://cran.r-project.org/web/packages/genSurv/index.html
[Accessed 9 11 2023].
[18] Daniela Dunkler, 2023. *Package 'shrink'.* [Online]
Available at: https://cran.r-project.org/web/packages/shrink/shrink.pdf
[Accessed 9 11 2029].
[19] Paul Christian Bürkner, 2023. *Package 'brms'.* [Online]
Available at: https://cran.r-project.org/web/packages/brms/brms.pdf
[Accessed 9 11 2023].
[20] Frank Harrell et al, 1996. Multivariable Prognostic Models: Issue in Developing Models, Evaluating Assumptions and Adequacy, and Measure and Reducing Errors. *Statistics in Medicine* 15(4), p361-387.
[21] Patrick Heagerty et al, 2005. Survival Model Predictive Accuracy and ROC Curves. *Biometrics* 61(1), p92-105.
[22] Ben Van Calster et al, 2019. Calibration: the Achilles heel of predictive analytics. *BMC Medicine* 17(230).