MSR109

Fitting flexible survival models to landmark survival estimates in R

Lydia Walder MSc[1], Robert Hettle MMath[2]

[1]Health Technology Assessment and Modelling Science, AstraZeneca, Barcelona, Spain, [2]Health Technology Assessment and Modelling Science, AstraZeneca, Cambridge, United Kingdom

Objectives

Parametric survival models can predict lifetime outcomes for cost-effectiveness studies. Models typically are fitted to individual-patient data (IPD) using maximum likelihood estimation.

Where IPD are unavailable, e.g. for early cost-effectiveness studies, models can be fitted to landmark survival estimates.

An existing tool, SurvInt (Gallacher 2024)[1], has focused on standard models with simple hazard functions, which may not adequately represent the complex hazards observed in oncology.

Here, we expand on the existing methods by introducing an open-source function for fitting flexible Royston-Parmar (RP) cubic-spline models to landmark survival estimates.[2]

Methods

An R function, *solve_flexible_coeffs*, was developed to fit RP cubic-spline models to a vector of user-specified landmark survival estimates (see function below). Using *optim*[3] and *flexsurvspline*[4] RP spline functions, the function identifies the set of parameters that minimise the root mean squared error (RMSE) of the model predictions versus the landmark survival inputs.

To test the function, we fitted RP cubic-spline models to landmark survival estimates extracted from a publicly available dataset (cancertrials.io)[5] and compared survival projections to the *flexsurvspline*[4] RP cubic-spline models fitted to pseudo IPD from the same dataset.

Survival extrapolations were compared on a single plot, using *geom_ribbon* to represent the range (maximum – minimum curve) of projections from each.

```
Main function 1
solve_flexible_coeffs <-
function(lands,times,knots_s,scale_s){
max_time <- max(times)
min_time <- 0.00001

#Setting up the variables for the knot
positioning, knots are evenly distributed along
the log scale of the time vector
k_n_1 <- 1
k_n_adjusted_1 <- k_n_1 + 2
k_n_2 <- 2
k_n_adjusted_2 <- k_n_2 + 2
k_n_3 <- 3
k_n_adjusted_3 <- k_n_3 + 2

if(knots_s <=0 | knots_s > 3) #Test to make sure
the user is inputting enough data for a solution
to be found
{print("NUMBER OF KNOTS SHOULD BE DEFINED AS
1,2,OR 3. PLEASE UPDATE YOUR INPUTS")} else{

#Define knot locations
if(knots_s == 1){
knots_location <-
seq(from=log(min_time),to=log(max_time),length.ou
t = k_n_adjusted_1)
}else if (knots_s == 2){
knots_location <-
seq(from=log(min_time),to=log(max_time),length.ou
t = k_n_adjusted_2)
}else{knots_location <-
seq(from=log(min_time),to=log(max_time),length.ou
t = k_n_adjusted_3)}

#Using optim for the flexible spline distribution
parameters
initial_params <- rep(0, times = knots_s+2)
params <- stats::optim(
par = initial_params, #Initial parameter guess
fn = solve_flexible_fun, #Objective function
lands_f=lands,times_f=times,knots_f=knots_locatio
n,scale_f=scale_s, #Set landmarks, times, knot
location and scale
method = "Nelder-Mead" #Optimization method)
```

```
#Returning the outputs
output <-  list()
output <- list(parameters = params$par,
knot_locations = knots_location)
return(output)}}

Sub function, used within the main function
solve_flexible_fun <-
function(x,times_f,lands_f,scale_f,knots_f) {
#Calculate the survival function of the flexible
distribution at a vector of time values
preds <- 1 - flexsurv::psurvspline(q=times_f,
gamma=x, knots=knots_f,scale=scale_f)
#Calculate the errors
error <- (abs(lands_f-preds))^2
#Calculate RMSE
rmse <- sqrt(mean(error))
return(rmse)}
```

Results

The function successfully minimised the RMSE (<0.03) for the example dataset, predicting curves that aligned to the user-defined landmark survival estimates extracted from the observed Kaplan-Meier data.

The predictions performed well when compared to the *flexsurvspline* models fitted to the IPD. Deviation in the tails of the predicted and IPD curves was present, and the predictions were sensitive to the position of the final landmark timepoint. However, there was substantial overlap across the two methods.

Example: Non-small cell lung cancer (NSCLC), pembrolizumab-chemotherapy, progression-free survival[5,6]

Function performance

- Figure 1 highlights the function outputs in the form of 12 flexible survival curves. The curves accurately reflected the landmark survival estimates.
- Figure 2 presents the *geom_ribbons* generated from *solve_flexible_coeffs* and *flexsurvspline,* and the best statistically fitting curve according to AIC (informed by *flexsurvspline*).
- The tail of the *geom_ribbon* widened more for the *flexsurvspline* outputs, compared to *solve_flexible_coeffs.*
- The *geom_ribbon* from *solve_flexible_coeffs*, fell almost entirely within the *geom_ribbon* from *flexsurvspline*. The consistency in the statistically best fitting curve illustrates the alignment of the two methods.

Removing the final landmark survival estimate

- Our function places universal weight across all the landmark survival estimates, whereas the *flexsurvspline* function places greater weight to periods with a higher density of events.
- This behavior reduces the variation of the curves from *solve_flexible_coeffs*, compared to the *flexsurvspline* outputs (Figure 2).
- By removing the final landmark survival estimate, we can increase the variation in the tails of the curves (Figure 3) and potentially achieve more accurate long-term predictions.
- Future work could formalise this approach by including weights in the error minimisation step, to alter the influence of the individual landmarks in the model predictions.

Figure 1: Outputs from *solve_flexible_coeffs*

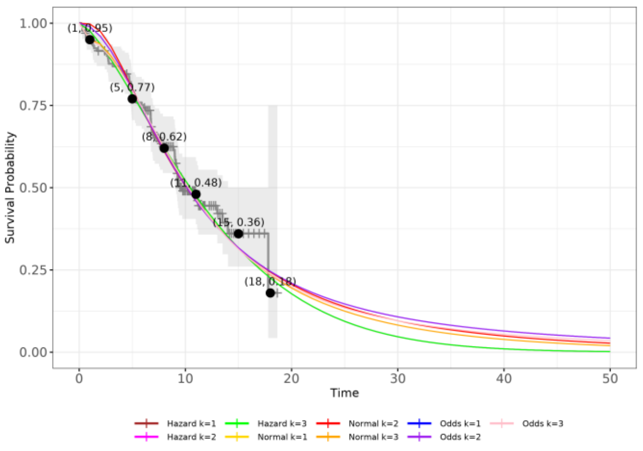


Figure 2: Comparison of *solve_flexible_coeffs* and *flexsurvspline*

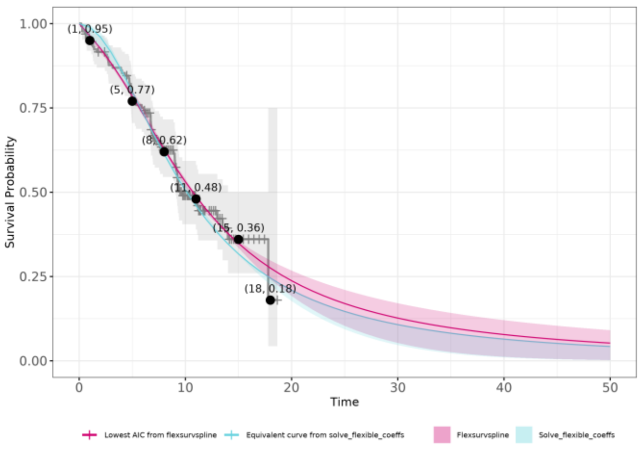
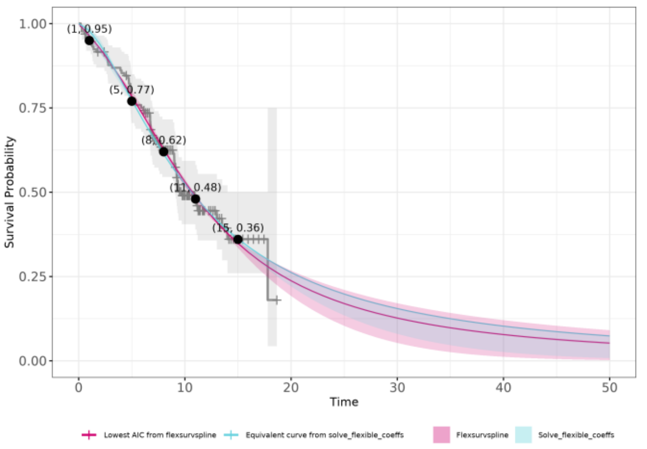


Figure 3: Removal of the final landmark survival estimate



Conclusions

- In testing the function successfully minimised the RMSE across for the example dataset and produced survival curves that align with *flexsurvspline* outputs, validating its performance.
- Limitations include the universally weight attributed to all landmark estimates. Hence the function is sensitive to the choice of user defined inputs, particularly at the later timepoints. Future work could consider including weights in the error minimisation step.
- Overall, in the absence of IPD, this function offers users the ability to capture more complex survival functions, reducing the need to rely on simpler models which may fail to capture the true survival trajectory for oncology diseases.

References

1. Gallacher. SurvInt: a simple tool to obtain precise parametric survival extrapolations. BMC Med Inform Decis. Mak. 2024
2. Royston and Parmar. Flexible parametric proportional-hazards and proportional-odds models for censored survival data, with application to prognostic modelling and estimation of treatment effects. 2002. Statistics in Medicine
3. *optim*. https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/optim
4. *flexsurvspline*. https://www.rdocumentation.org/packages/flexsurv/versions/2.3.2/topics/flexsurvspline
5. Palmer A et al. Combinatorial benefit without synergy in recent clinical trials of immune checkpoint inhibitors. 2020. medRxiv
6. Gandhi, L. et al. Pembrolizumab plus Chemotherapy in Metastatic Non-Small-Cell Lung Cancer. 2018. N. Engl. J. Med.

Supported by AstraZeneca

Presented at the ISPOR Europe 2025, Glasgow, Scotland