GLW Package

Abstract

The **glw** package is an environment made to analyse data (censored or not) through regression models considering a finite mixture where the comoponents are the Gamma, Lognormal and Weibull densities, the GLW mixture.

Keywords: Finite mixtures, Censored data, Bayesian estimation, R.

1. The glw finite mixture

The GLW mixture is a finite mixture with three components, the densities from the Gamma, Lognormal and Weibull distributions, which are written in terms of the mean, μ, and the variance, σ^2. Thus, the densiity function of the GLW model is

$$\begin{aligned} f(t_i|\boldsymbol{\theta}) &= p_1 f_1(t_i|\mu,\sigma) + p_2 f_2(t_i|\mu,\sigma) + p_3 f_3(t_i|\mu,\sigma) \\ &= \sum_{j=1}^{3} p_j f_j(t_i|\mu,\sigma)\ , \quad t_i > 0, \end{aligned} \tag{1}$$

where $\boldsymbol{\theta} = (\mu, \sigma, \mathbf{p})$, $p_j > 0$, $j = 1, 2, 3$ and $\sum_{j=1}^{3} = 1$ are the mixture weights, $f_1(.|\mu,\sigma)$, $f_2(.|\mu,\sigma)$ and $f_3(.|\mu,\sigma)$ are the densities of the Gamma, Lognormal and Weibull distributions, respectively.

Similarly, the survival function is

$$S(y|\boldsymbol{\theta}) = p_1 S_1(y|(\mu,\sigma)) + p_2 S_2(y|(\mu,\sigma)) + p_3 S_3(y|(\mu,\sigma)), \tag{2}$$

where $S_j(.|.)$, is the survival function associated with the density $f_j(.|.)$, $j = 1, 2, 3$.

To acommodate censored data, we insert the following notation:

- T_i - survival time;
- $(L_i, R_i]$ - censorship interval;
- $Y_i = max\{min\{T_i, R_i\}, L_i\}$ - observed time;
- $\boldsymbol{\delta}_i = (\mathbb{I}(Y_i = T_i), \mathbb{I}(Y_i = R_i), \mathbb{I}(Y_i = L_i))$ - censorship indicator.

The vectors $\boldsymbol{\delta}_i$, $i = 1, \ldots, n$, assume the value $(0, 0, 1)$ if the i^{th} observation is left censored, $(0, 1, 0)$ when it is right censored, $(1, 0, 0)$ when y_i is a survival time and $(0, 0, 0)$ when it is interval censored, case in which we record the interval $y_i = (l_i, r_i]$. If $L_i = 0$, then $Y_i =$

$min\{T_i, R_i\}$ is a survival or right censored time. When $R_i = \infty$, then $Y_i = max\{T_i, L_i\}$ is a survival or left censored time. The observed data is $D = (\mathbf{y}, \boldsymbol{\delta}) = \{(y_1, \boldsymbol{\delta}_1), \ldots, (y_n, \boldsymbol{\delta}_n)\}$.

Assuming that the censorship is non-informative and independence between censor and survival times (**?**) the likelihood generated by D is

$$L(\boldsymbol{\theta}|D) = \prod_{i=1}^{n} f(y_i, \boldsymbol{\delta}_i|\boldsymbol{\theta}) \tag{3}$$

$$\propto \prod_{i=1}^{n} \Big(\sum_{j=1}^{3} p_j f_j(y_i) \Big)^{\delta_{i1}} \Big(\sum_{j=1}^{3} p_j S_j(y_i) \Big)^{\delta_{i2}} \Big(\sum_{j=1}^{3} p_j F_j(y_i) \Big)^{\delta_{i3}} \Big(\sum_{j=1}^{3} p_j \Big(F_j(r_i) - F_j(l_i) \Big) \Big)^{1-\sum_l \delta_{il}}$$

where $S_j(.)$ and $F_j(.)$ are the survival and cumulative distribution functions associated to the densities $f_j(.)$ (we consider $f_j(.) = f_j(.|\mu, \sigma)$), $j = 1, 2, 3$.

We insert covariates in the model using a linear predictor with a logarithmic link function on the mean through the equation

$$log(\mu_i) = \mathbf{x}_i^\top \boldsymbol{\beta} = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_r x_{ir} \ , \ i = 1, \ldots, n, \tag{4}$$

where $\mathbf{x}_i^\top = (1, x_{i1}, \ldots, x_{ir})$ is the vector of covariates for the i^{th} observation. In order to obtain a more flexible, yet parsimonious, model we allow each mixture component to have its own intercept β_0^j. Thereby, we have

$$log(\mu_{ij}) = \beta_0^{(j)} + \beta_1 x_{i1} + \ldots + \beta_r x_{ir} \ , i = 1, \ldots, n, \tag{5}$$

where the index j refers to the mixture components.

1.1. Prior and Posterior Distributions

The likelihood of the GLW mixture is a sum with 3^n terms which means that there are no conjugated families (**?**). So, we may consider any distribution with an adequated support to the parameters involved. The default prior distributions in the **glw** package are

- $\boldsymbol{\beta} \sim Normal_q(\mathbf{0}, 10^2 \mathbb{I})$;
- $\sigma^2 \sim Gamma(0.01, 0.01)$;
- $\mathbf{p} \sim Dir(1, 1, 1)$;

where the subscript q is the dimension of the vector $\boldsymbol{\beta}$ and $\mathbb{I}$ is the identity matrix.

The posterior distribution is given by:

$$\begin{aligned} \pi(\boldsymbol{\theta}|D) &\propto L(\boldsymbol{\theta}|D)\pi(\boldsymbol{\theta}) \\ &\propto \prod_{i=1}^{n} \Big\{ f(y_i, \delta_i|\boldsymbol{\beta}, \sigma, \mathbf{p}) \Big\} \pi(\boldsymbol{\beta})\pi(\sigma^2)\pi(\mathbf{p}), \end{aligned}$$

which is not known. So, the estimation procedure is based on MCMC samples generated from this posterior distribution through the *Adaptive Metropolis* algorithm **?**.

1.2. Cure Rate Models

In the usual Survival Analysis models, we assume that the whole population is susceptible to the ocurrence of the event of interest. But, sometimes, this might not be the case. For example, when studying cancer recurrence time, a fraction of the population may never have cancer again. In scenarios like this we consider the cure rate (or long-term) models. In this Section we briefly describe the Standard Cure rate model (**?**) and the Promotion Time model (**?**).

Standard Cure Rate Model

For the Standard cure rate model, we consider the (improper) populational survival function

$$S_{pop}(y|\boldsymbol{\psi}) = \pi + (1-\pi)S(y|\boldsymbol{\theta}) \ , \ y \in \mathcal{R}, \ \pi \in (0,1) \ , \tag{6}$$

where $\boldsymbol{\psi} = (\pi, \theta)$ is the proportion of cured patients and $S(.|\boldsymbol{\theta})$ is the survival function of the individuals that are susceptible to the event of interest.

Thus, the distribution of the observed times $\mathbf{Y}$ and the censorship indicator $\boldsymbol{\delta}$ ($\delta_i = 1$ if the i^{th} observation is a failure time and $\delta_i = 1$ if y_i is a right censored time) is

$$\begin{aligned} f_{pop}(y, \delta|\boldsymbol{\psi}) &\propto \Big(f_{pop}(y)|\boldsymbol{\psi}\Big)^{\delta}\Big(S_{pop}(y)|\boldsymbol{\psi}\Big)^{1-\delta} \\ &= \left((1-\pi)\sum_{j=1}^{3} p_j f_j\Big(y|(\mu,\sigma)\Big)\right)^{\delta}\left(\pi + (1-\pi)\sum_{j=1}^{3} p_j S_j\Big(y|(\mu,\sigma)\Big)\right)^{1-\delta} \end{aligned} \tag{7}$$

The covariates are incorporated in the model through the parameter π using a *logit* link function

$$g(\pi_i) = log\left(\frac{\pi_i}{1-\pi_i}\right) = \mathbf{x}_i^{\top}\boldsymbol{\beta} = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_r x_{ir} \ , \ i = 1, \ldots, n, \tag{8}$$

where $\mathbf{x}_i^{\top} = (1, x_{i1}, \ldots, x_{ir})$ is the covariates vector for the i^{th} observation and $\boldsymbol{\beta}^{\top} = (\beta_0, \beta_1, \ldots, \beta_r)$ are the regression coefficients.

Finally, the likelihood of the standard cure rate model, generated by $(\mathbf{y}, \boldsymbol{\delta}) = \big((y_1, \delta_1), \ldots, (y_n, \delta_n)\big)$, is

$$\begin{aligned} L(\boldsymbol{\psi}|(\mathbf{y}, \boldsymbol{\delta})) &= \prod_{i=1}^{n} f_{pop}(y_i, \delta_i|\boldsymbol{\theta}) \\ &\propto \prod_{i=1}^{n}\left((1-\pi_i)\sum_{j=1}^{3} p_j f_j\big(y_i|(\mu,\sigma)\big)\right)^{\delta_i}\left(\pi_i + (1-\pi_i)\sum_{j=1}^{3} p_j S_j\big(y_i|(\mu,\sigma)\big)\right)^{1-\delta_i} . \end{aligned}$$

Promotion Time Cure Rate Model

The populational survival function in the promotion time cure rate model is

$$S_{pop}(y|\boldsymbol{\psi}) = e^{-\eta F(y|\boldsymbol{\theta})} \ , \ y \in \mathcal{R}, \ \eta > 0 \ , \tag{9}$$

where $F(.|.)$ is the cdf of the susceptible individuals and the the proportion of cured is $e^{-\eta}$. The covariates are incorporated in the model through the parameter η using a *log* link function

$$log(\eta_i) = \mathbf{x}_i^\top \boldsymbol{\beta} = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_r x_{ir} \ , \ i = 1, \ldots, n. \tag{10}$$

Similarly to the standard model, it can be shown that the likelihood for the promotion time model is

$$\begin{aligned} L(\boldsymbol{\psi}|(\mathbf{y}, \boldsymbol{\delta})) &= \prod_{i=1}^{n} f_{pop}(y_i, \delta_i|\boldsymbol{\theta}) \\ &\propto \prod_{i=1}^{n} \left(\eta_i f(y_i|\boldsymbol{\theta}) e^{-\eta_i F(y_i|\boldsymbol{\theta})}\right)^{\delta_i} \left(e^{-\eta_i F(y_i|\boldsymbol{\theta})}\right)^{1-\delta_i} . \end{aligned} \tag{11}$$

1.3. Hypothesis Testing

To assess the effect of the covariates we can test hypothesis of the form

$$\begin{cases} H : \mathbb{C}\boldsymbol{\beta} = \mathbf{0} \\ A : \mathbb{C}\boldsymbol{\beta} \neq \mathbf{0} \end{cases} \tag{12}$$

where $\mathbb{C}$ is a contrast matrix.

We can also evaluate the need for the three components of the GLW mixture by performing hypothesis tests on the mixture weights. In this case, we consider the hypotheses

$$\begin{cases} H : p_j = 1 \quad (\Leftrightarrow p_i = 0 \quad \forall\, i \neq j) \\ A : p_j < 1 \end{cases} \quad \text{and} \quad \begin{cases} H : p_j = 0 \\ A : p_j > 0 \end{cases} . \tag{13}$$

The null hypotheses given by (12) and (13) are sharp hypotheses (**?**). So, to perform tests under the Bayesian framework, we consider the Full Bayesian Significance Test (FBST) **?**, which gives us an evidence in favor of the null hypothesis H. The FBST has two steps. In the first step we must find the posterior maximum under the null hypothesis H, which involves an optimization procedure and is computed using the function `mode_glw(...)`. The second step is an integration, wich is made using the MCMC sample.

2. The glw package

The estimation of the parameters of the models described in the previous Section is made through samples of their posterior distributions generated via MCMC simulation. So, in order to fit these models to some data, we constructed the **glw** package, which allows us to estimate the parameters, the survival curves, the predictive distributions, compute measures of model adequability to the data and perform hypothesis tests. We also developes some functions to generate data from the proposed models and we use the **LaplacesDemon** package (**?**) to generate samples from the posterior distributions of the parameters.

The following functions allow us to compute the density, the cdf and generate data from the GLW mixture.

- `dglw(x, P, Mu, S2)` returns the density of the GLW mixtue;
- `pglw(x, P, Mu, S2, lower.tail=T)` returns the cdf of the GLW mixtue;
- `rglw(n, P, Mu, S2)` generates a random sample of the GLW mixture.

The arguments of this functions are:

- `x` quantile
- `P` vector containing the three mixture weights;
- `Mu` mean;
- `S2` variance;
- `lower.tail` logical; if TRUE (default), probabilities are $P[X <= x]$, otherwise, $P[X > x]$;
- `n` sample size.

If we want to generate samples on a regression model framework with possible left or right censored data, or from the standard or promotion time cure rate models we consider, respectively, the functions

- `rglw3(n, Betas, X, P, Sigma, Pc = c(0, 0))`
- `rscr_glw(n, Mu, Sigma, P, Betas, X, pc)`
- `rpt_glw(n, Mu, Sigma, P, Betas, X, pc)`

where we have

- `n` sample size;
- `Betas` regression coefficients;
- `X` matrix of covariates (with first column of $1's$);
- `P` vector containing the three mixture weights;
- `Sigma` variance;
- `Pc` vector with probabiliity of right and left censored times.
- `Mu` mean for the time of the susceptible individuals;
- `pc` probablity of right censored times.

Examples

```
### Sample of size 10
> rglw(n=10, P=c(0.2, 0.3, 0.5), Mu=10, Sigma=10)
```

```
 [1] 13.490200  8.511048  6.556054  5.325755 13.226479 13.521355
 [7] 11.384301 17.026614 14.948169 11.447700

### Sample of size 10 with no covariates and 20% of right censored times
> rglw3(n=10, Betas=2, X=as.matrix(rep(1,10)), P=c(0.2, 0.3, 0.5), Sigma=10, Pc=c(0.2,
0))

               Y d
 [1,]  7.4264029 1 1
 [2,]  8.0049152 0 1
 [3,]  0.2135202 0 1
 [4,]  6.8530548 1 1
 [5,]  7.2614250 1 1
 [6,]  2.9179881 0 1
 [7,]  8.8628923 1 1
 [8,]  6.5729715 0 1
 [9,]  3.4388514 1 1
[10,] 10.5202133 1 1

### Sample of Standard cure rate model with two groups with 30% and 52% of cured
and 10% of right censored times
### This data will be analyzed to exemplify the other functions along this tutorial
> data <- rscr_glw(n=100, Mu=20, Sigma=10, P=c(0.2,0.3,0.5), Betas=c(log(0.3/0.7),
1), X=cbind(1, rbinom(100,1,0.5)), pc=0.10)
head(data)

          Y D X[1] X[2] W
1 24.314015 1    1    1 0
2 25.223561 1    1    1 0
3  2.258646 0    1    1 1
4 94.651474 0    1    1 1
5 21.504615 1    1    0 0
6 15.316438 1    1    0 0
```

2.1. Fitting the GLW mixture to data

To fit the glw mixture we consider three main comands `glwfm(...)`, `glw_scr(...)` and `glw_ptcr(...)`for the models in (2), (6) and (9). They have the same arguments:

- `parm0`: is the list of inital values. For the main model the order is: the first J values are the initial values of the regression coefficients, where J is the dimension of $\boldsymbol{\beta}$. The value at position $J+1$ is for the variance σ^2 and the remaining values are for the mixture weigths $\mathbf{P} = (P_1, P_2, P_3)$. On the cure rate models, the order is: the first position is the initial value for the mean μ, the second is for the variance σ^2, the next positions are for the weights and the last positions are for the regression coefficients (related to the cure rate fraction).
- `Data`: is a list containing

  - `y`: the observed times.
  - `d`: the censorship indicator.
  - `yf`: the end of the interval censored times
  - `y`: the observed times
  - `X`: matrix of covariates (model matrix). Vector of 1's when tere are no covariates.
  - `rint`: logical; True or False indicating when to consider different intercepts (model 5) or not (model 4).

- `Specs`: is a list containing some of the simulation parameters
  - `Iterations`: total size of the MCMC generated sample.
  - `Status`: number of Iterations that the simulations status will be printed on the console.
  - `Thinning`: thinning of the generated sample.
  - `Algorithm`: name of algorithm used in the generation of the MCMC sample. Usually `Algorithm='AM'` for Adaptive Metropolis.
  - `LogFile`: a path to where the log of the simulation should be printed

- `Specs2`: is a list containing simulation parameters specific to the chosen algorithm. For 'AM' algorith, the default is `Specs2 = list(Adaptive = 1000, Periodicity = 100)`.

- `prior`: a list of the log of the prior distributions, except for the mixture weights, where we chose the values of the hyperparameter α from a Dirichlet distribution. The default is `prior=list(beta='dnormv(beta, 0, 100, log=T)', sigma='dexp(sigma, 1, log=T)', alpha==c(1,1,1))`. for the main model. For the cure rate model we may add the prior for the parameter μ.

- `mixture`: the mixture to be fitted. Choices are
  - `'glw'` for a Gamma-Lognormal-Weibull finite mixture;
  - `'gl'` for a Gamma-Lognormal finite mixture;
  - `'gw'` for a Gamma-Weibull finite mixture;
  - `'lw'` for a Lognormal-Weibull finite mixture;
  - `'g'` for a Gamma model;
  - `'l'` for a Lognormal model;
  - `'w'` for a Weibull model;

Examples

```
### Standard cure rate model fitted to simulated data,
### The generated sample is stored in glw_scr(...)$Posterior1
> N <- nrow(data)
```

```
> Y <- data$Y
> D <- data$D
> X <- data[,3:4]

> MyData <- list(y=Y, d=D, X=X)
> Specs <- list(Iterations=10**4, Status=10**3, Thinning=1, Algorithm="AM", LogFile='')
> prior <- list(beta='dnormv(beta, 0, 100, log=T)', mu='dgamma(sigma, 0.01, 0.01,
log=T)', sigma='dgamma(sigma, 0.01, 0.01, log=T)')
> parm0 <- c(1, 1, rep(1/3,3), rep(0, ncol(X)))

> Fit <- glw_scr(parm0=parm0, Data=MyData, Specs=Specs, prior=prior, mixture='glw')

> mcmc <- Fit$Posterior1
> round(colMeans(mcmc),2)

     mu   sigma    p[1]    p[2]    p[3] beta[1] beta[2]
  20.17   10.46    0.32    0.27    0.41   -1.38    0.36
```

2.2. CPO functions

We can fit several models when working with the GLW mixture. We can use the LPML to find which model is best for the data, but first we need to compute the CPO. We have three different functions `cpo_glw` for the main model, `cpo_scr` for the standard cure rate model and `cpo_ptcr` for the promotion time model. These three functions have the same arguments

- `parm`: matrix; a sample from the posterior distribution of the parameters, obtained by `Fit$Posterior1`.
- `Data`: list; the same list used to fit the model.
- `mixture`: string; indicates wich mixture you considered in fitting the model.

Continuing the example, we compute the cpo and find the lpml as the sum of the logarithm of the cpo's

```
### Computing the CPO for the standard cure rate model adjusted to data
cpo <- cpo_scr(mcmc, MyData, mixture='glw')
lpml <- sum(log(cpo)) ; lpml

[1] -189.1265
```

2.3. Hypothesis testing

As mentioned before, to perform the FBST we need an optimization step. This is done in the **glw** package throught the functions `mode_glw` for the main model, `mode_scr` and `mode_pt` for the standard and promotion time cure rate models. The arguments of these functions are: `Data`, the same list used to fit the model plus the vector of inital values `initial.par`,

`prior`, a list of the prior distributions, and `mixture`, a string indicating the mixture (same as before). These functions return a list, which contains the posterior mode, the value of the log-posterior evaluated at the mode, an indicator of convergence and a message, which come from the `optim` function.

In the example, we can asses the covariate effect testing the hypothesis $H : \beta_1 = 0$. To compute the FBST evidence in favor of H, first we must find the maximum of the (log) posterior under the null hypothesis, which is done using the following code

```
> fbst.data <- list(y=Y, d=D, X=rep(1,N), initial.par=colMeans(mcmc)[1:6])
> mode <- mode_scr(fbst.data, prior, mixture='glw') ; mode

$Posterior.mode
     mu   sigma    p[1]    p[2]    p[3] beta[1]
19.9654  9.5483  1.0000  0.0000  0.0000 -0.9884

$value
[1] -202.5846

$Convergence
[1] 0

$message
NULL
```

To compute the evidence we need the value of the log-posterior of each observation generated with the MCMC simulation. These values are stored in `Fit$Monitor`.

```
LP <- Fit$Monitor #Log-Posterior
1 - mean(LP>mode$value)

[1] 1
```

2.4. Estimating survival curves

Similarly to the CPO and mode computations, there is also three functions to estimate the survival functions after we fitted the model. The first one is `surv_glw(...)` and, for the cure rate models we have `Spop_scr(...)` and `Spop_ptcr(...)`. These three functions have the same parameters:

- `parm`: matrix; a sample from the posterior distribution of the parameters, obtained by `Fit$Posterior1`
- `y`: a vector of observed times
- `X`: a vector containing the covariates for one observation
- `mixture`: string indicating the mixture fitted to the data. For the `surv_glw` functions the options are only the mixtures ('glw', 'gl', 'gw', 'lw'). For the other two functions we can also choose the Gamma, Lognormal or Weibull models ('g', 'l', 'w').

- by: increment of time

These functions return a list containing two objects: $surv contains the estimated survival probabilities and $time contains the respective times. To estimate the survival times for the two groups in our simulated data we need the following commands

```
### Estimating the survival function
> S0 <- Spop_scr(mcmc, y=Y, X=c(1,0), mixture='glw', by=0.01)
> S1 <- Spop_scr(mcmc, y=Y, X=c(1,1), mixture='glw', by=0.01)

> s0 <- S0$surv ; s1 <- S1$surv ### Survival Probabilities
> t0 <- S0$time ; t1 <- S1$time ### Time

> plot.data <- data.frame(s0, s1, t0, t1) > plot <- ggplot(plot.data, aes(x=t0,
y=s0)) + geom_line(size=1.05) + xlab('Time') + ylab('Estimated Survival') + ylim(0,1)
+ xlim(10,50) + geom_line(aes(x=t1, y=s1), size=1.05, col=2) ; plot
```

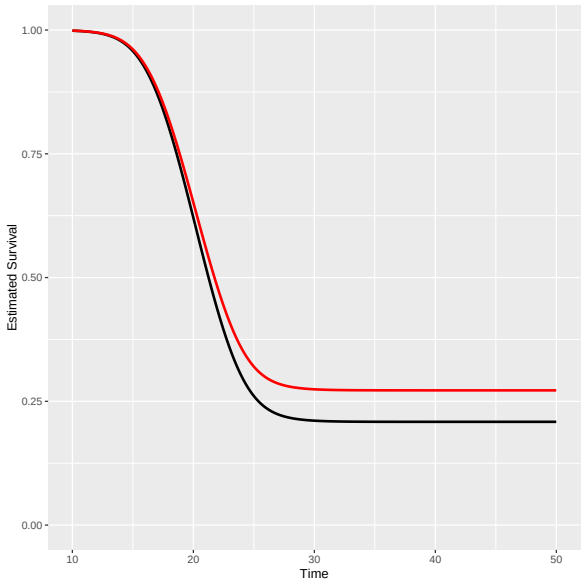

Figure 1: Estimated survival probabilities for the two groups

References

Affiliation:

Firstname Lastname
Affiliation
Address, Country
E-mail: name@address
URL: http://link/to/webpage/