Prediction in Joint Models

1 Description

This method is the Pyhton adaptation of R's survfitJM function. It computes $\pi(s+t|s)$ the probability at time s of surviving over time s+t.

The probability for a subject i we know alive at time s to survive over time $s+t$ can be formulated by the following formula :

$$\pi_i(s+t|s) = \mathbb{P}(T_i^* \geq s+t | T_i^* > s, \mathcal{Y}_i(s); \theta)$$

$$= \int \frac{S_i(s+t|\mathcal{M}_i(s+t, b_i, \theta); \theta)}{S_i(s|\mathcal{M}_i(s, b_i, \theta); \theta)} * p(b_i|T_i^* > s, \mathcal{Y}_i(s); \theta) db_i$$

Where the different terms denotes :

- $\pi_i(x)$: Probability for subject i to be alive at time x
- s : time from when we know or we assume the subject is alive and from when we want to compute predictions of his future survival probabilities
- t : time horizon from s. The survival probability is given for time $s+t$
- T_i^* : Random variable representing time when subject i dies
- $\mathcal{Y}_i(x)$: Longitudinal measurement of subject i before time x
- θ : Parameters of joint model
- $S_i()$: Survival function of subject i
- $\mathcal{M}_i()$: Longitudinal history of subject i, approximated by the linear mixed-effects model
- b_i : Subject i random effects

2 Usage

Call .Survfit()

Ones your JointModel object is fitted, call *object.Survfit(new_data, id_var)*. *new_data* and *id_var* are the only required arguments. The other arguments are optional.

Default values

- *surv_times* = None
- *last_time* = None
- *ci* = numpy.array([0.025, 0.975])
- M = 200
- *scale* = 1.6
- *simulate* = False

3 Arguments

- *new_data* : A pandas dataframe containing covariates used in both survival and linear mixed-effects models and longitudinal information ordered by increasing time for each subject. *new_data* must also containing a column that identifies different subjects. The names of covariates columns must be the same in *new_data* and in data used to fit the model. This dataframe is structured with one line for each longitudinal information. It could contains informations for several subjects. An exemple of a valid new_data is provided just below

id_subject	longitudinal var	Linear var 1	Surv var 1	Surv var 2
1	221	12	58	0
1	257	24	58	0
1	284	36	58	0
2	112	14	65	1
2	191	26	65	1

- *id_var* : Name of the column that identifies subjects in *new_data*.

- *surv_times* : Numerical numpy array containing one or several times $s + t$ of predictions. If surv_times is None, $s + t$ will be automaticaly generated.

- *last_time* : s time from when we know or assume a subject is alive and from when we want to predict at time $s + t$. *last_time* could be a character string or a numeric numpy array. If *last_time* is a string, the name of a column in *new_data* containing s time from which we predict is expected in input. If *last_time* is a numpy array, it must be a vector containing s time for each subject. If *last_time* is None, last longitudinal time in *new_data* will be taken as *last_time*. **Warning** each subject must have only one s time.

- *ci* : Numerical numpy array that specifies which quantiles to use for the computation of confidence interval for the predicted probabilities.

- M : Integer denoting how many loop are computed in Monte-Carlo method to estimate survival probabilities and compute a confidence interval.
- *scale* : A numeric scalar that controls the acceptance rate of the Metropolis-Hastings algorithm
- *simulate* : A boolean (True or False) that specifies if we estimate our survival probabilities using Monte-Carlo method or not. If *simulate* is True, survival probabilities will be computed by a Monte-Carlo method and a confidence interval will be provided. If *simulate* is False, probabilities will be computed without Monte-Carlo method and only ponctual estimation will be return.

4 Details

Estimation of $\pi_i(s+t|s)$ computation method will depend on *simulate* argument.

simulate = True Estimation will be based on following Monte-Carlo procedure :

Step1 : Simulate $\theta^{(l)}$ vector of parameters values from a multivariate normal distribution $\mathcal{N}(\hat{\theta}, C(\hat{\theta}))$ where $\hat{\theta}$ are the fitted joint model's parameters estimated by MLE and $C(\hat{\theta})$ their variance-covariance matrix.

Step2 : Simulate $b_i^{(l)}$ random effects of subject i from b_i posterior distribution given $T_i^* > s$, $\mathcal{Y}_i(s)$ and $\theta^{(l)}$. This is achieved using a Metropolis-Hastings algorithm with independent proposals from a properly centered and scaled multivariate t distribution. The *scale* argument controls the acceptance rate for this algorithm.

Step3 : Compute:

$$\pi_i^{(l)}(s+t|s) = \frac{S_i(s+t|\mathcal{M}_i(s+t, b_i^{(l)}, \theta^{(l)}); \theta^{(l)})}{S_i(s|\mathcal{M}_i(s, b_i^{(l)}, \theta^{(l)}); \theta^{(l)})}$$

Steps 1-3 are repeated $l = 1, ..., M$ times M is given by M argument of *Survfit()* method.

simulate = False Survival probabilities will be estimated by :

$$\tilde{\pi}_i(s+t|s) = \frac{S_i(s+t|\mathcal{M}_i(s+t, \hat{b}_i^{(s)}, \hat{\theta}); \hat{\theta})}{S_i(s|\mathcal{M}_i(s, \hat{b}_i^{(s)}, \hat{\theta}); \hat{\theta})}$$

Where the different terms denotes :

- $\tilde{\pi}_i(x)$: Estimated probability for subject i to be alive at time x

- s : time from when we know or we assume the subject is alive and from when we want to compute predictions of his future survival probabilities
- t : time horizon from s. The survival probability is given for time $s+t$
- $\hat{\theta}$: MLE of parameters of joint model
- $\hat{b}_i^{(s)}$: Mode of the conditional distribution $p(b_i|T_i^* > s, \mathcal{Y}_i(s); \hat{\theta})$
- $S_i()$: Survival function of subject i
- $\mathcal{M}_i()$: Longitudinal history of subject i, approximated by the linear mixed-effects model

5 Value

A dictionary containing a pandas dataframe for each group inputed in *new_data*. Each dataframe provide estimated probabilities to survive at each $s+t$ times. If *simulate* is True, the returned dataframe will contain a summary of M predictions containing : mean, median, low and high boundaries of confidence interval. And if *simulate* is False, only ponctual estimation will be returned.

6 References

Rizopoulos, D. (2012) *Joint Models for Longitudinal and Time-to-Event Data: with Applications in R.* Boca Raton: Chapman and Hall/CRC.