`RecForest` : Forêts aléatoires de survie pour l'analyse des événements récurrents en R

Juliette Murris* Sandrine Katsahian† Audrey Lavenu‡

Résumé

Les forêts aléatoires de survie permettent de modéliser les effets de prédicteurs sur des données de survie en s'affranchissant d'hypothèses, comme la linéarité ou la faible dimension (le nombre d'individus supérieur au nombre de prédicteurs). Leur utilisation est ainsi accrue en recherche médicale. Nous avons récemment adapté ces forêts aux données de survie avec événements récurrents avec `RecForest`, en présence ou non d'un événement terminal pour prédire le nombre attendu d'événements. L'objectif de ce travail est d'introduire l'utilisation de `RecForest` en programmation R. Notre approche est en 5 étapes, pour i) discerner la pertinence des événements récurrents et terminaux, ii) développer des arbres et construire la forêt, iii) évaluer minutieusement la performance, iv) fournir l'importance des variables, et v) permettre des prédictions sur de nouvelles données. Pour l'illustration, le jeu de données `readmission` du package `frailtypack` de R a été utilisé.
Mots-clefs : Biostatistique – Analyse de survie – Santé

1 Introduction

Les modèles d'apprentissage automatique, tels que les forêts aléatoires, sont de plus en plus appliqués à l'analyse de données de survie, notamment avec l'utilisation courante des forêts aléatoires de survie (RSF) [Huang et al., 2023, Ishwaran et al., 2008]. Toutefois, face à des événements récurrents tels que les hospitalisations, les rechutes, ou les crises d'asthme, l'analyse de survie classique, et ainsi les RSF, ne considère que la première occurrence. Pour pallier cela, nous avons étendu les RSF aux événements récurrents pour prédire le nombre attendu d'événements pour chaque individu au cours du temps avec `RecForest`. L'objectif ici est d'introduire l'utilisation de `RecForest` en programmation R.

2 Vue d'ensemble de l'algorithme

Les données en entrée sont consitutées de plusieurs observations pour chaque individu, avec des variables indiquant la présence (ou non) d'événements récurrents, la présence (ou non) d'un événement terminal, le temps associé à la survenue de chaque événement, et d'éventuelles variables explicatives, dépendantes ou indépendantes du temps.

`RecForest` fonctionne de manière similaire aux RSF de Ishwaran et al. [2008]. Pour chaque échantillon bootstrap, des arbres de survie adaptés aux événements récurrents sont construits. La forêt agrège les résultats obtenus pour chaque arbre afin d'obtenir une estimation globale.

Règle de division. A chaque noeud, l'objectif est de discriminer au mieux les données à partir de *mtry* variables sélectionnées aléatoirement. Cette discrimination est faite suivant la présence ou non d'un événement terminal en utilisant la statistique du pseudo-score test ou la statistique du test de Wald d'un modèle marginal de Gosh-Lin [Lawless and Nadeau, 1995, Ghosh and Lin, 2002].

Estimation et agrégation. L'estimation est le nombre attendu d'événements pour chaque individu jusqu'au temps t, notée $\hat{M}(t \mid x)$, qui est l'aggrégation des estimations $\hat{\mu}(t \mid \mathbf{x})$ pour chaque arbre de la forêt. Suivant la présence ou non d'un événement terminal,

$$\hat{M}(t \mid x) = \frac{1}{B}\sum_{b=1}^{B} \hat{\mu}_b(t \mid \mathbf{x}) = \begin{cases} = \frac{1}{B}\sum_{b=1}^{B} \hat{R}_b(t \mid \mathbf{x}) = \frac{1}{B}\sum_{b=1}^{B} \int_0^t \frac{N_b(du|\mathbf{x})}{Y_b(du|\mathbf{x})} & \text{sans événement terminal,} \\ = \frac{1}{B}\sum_{b=1}^{B} \int_0^t \hat{S}_b(u \mid \mathbf{x})\, d\hat{R}_b(u \mid \mathbf{x}) & \text{avec un événement terminal.} \end{cases}$$

*HeKA, Inria, Inserm, Université Paris Cité, Paris, France, `juliette.murris@inria.fr`
†Centre d'Investigation Clinique 1418 Épidémiologie Clinique, Paris, France, `sandrine.katsahian@aphp.fr`
‡Institut de Recherche Mathématique de Rennes (IRMAR), Rennes, France, `audrey.lavenu@univ-rennes.fr`

avec B le nombre d'arbres, $\mathbf{x}$ un vecteur de variables explicatives, dépendantes ou indépendantes du temps, N le nombre d'événements noeud-spécifique, Y le nombre d'individus à risque, $\hat{S}$ l'estimateur Kaplan-Meier de la fonction de survie.

Elagage de chaque arbre. De même que Devaux et al. [2023], nous proposons deux règles d'arrêt pour chaque nœud terminal : (i) un nombre minimal d'événements appelé *minsplit*, et (ii) un nombre minimal d'individus appelé *nodesize*.

Les sorties de l'algorithme correspondent aux prédictions du nombre attendu d'événements récurrents au cours du temps pour chaque individu.

3 Mise en pratique

Données d'entrée. Les données `readmission` sont fréquemment utilisées pour des principes méthodologiques [Rondeau et al., 2012]. Cette base contient les réhospitalisations de 403 patients atteint d'un cancer colorectal.

```
data(readmission, package = "frailtypack")
X <- readmission |>  dplyr::select(id, chemo, sex, dukes) |> dplyr::group_by(id) |>
    as.data.frame()
Y <- readmission |> dplyr::pull(id, t.start, t.stop, event, death)
```

Création d'un objet `RecForest`. Les hyper-paramètres à fixer (ou à optimiser) sont le nombre de variables tirées aléatoirement à chaque nœud est *mtry*, le nombre minimal d'événements est *minsplit*, le nombre minimal d'individus est *nodesize* et le nombre d'arbres *ntrees*.

```
mtry <- 5 # number of candidate variables randomly drawn at each node
minsplit <- 5 # minimal number of events required to split the node
nodesize <- 5 # minimal number of subjects required in both child nodes to split
n_trees <- 100 # number of trees
method <- "GL" # Ghosh-Lin for recurrent events, with a terminal event
params <- list(seed = seed, mtry = mtry, minsplit = minsplit, nodesize = nodesize,
    method = method, n_trees = n_trees)
my_recforest <- RecForest(X = X, Y = Y, params = params)
```

Evaluation. Pour l'évaluation des performances, nous utilisons des métriques adaptées aux événements récurrents, soient des versions étendues du C-index et de l'erreur quadratique moyenne.

```
c_index(my_recforest, X_new = NULL) # X_new = NULL refers to OOB samples
mse(my_recforest, X_new = NULL) # X_new = NULL refers to OOB samples
```

Importance des variables. L'importance d'une variable est évaluée par permutation, correspondant à l'impact de perturbations aléatoires dans l'échantillon sur l'erreur OOB.

```
vimp(my_recforest, n_permutations = 10)
```

Prédictions. A partir de nouvelles données en entrée, `RecForest` est utilisée pour prédire les nombres attendus d'événements pour chaque nouvel individu.

```
predictions = predict(my_recforest, X_new = X_new)
```

Références

Anthony Devaux, Catherine Helmer, Robin Genuer, and Cécile Proust-Lima. Random survival forests with multivariate longitudinal endogenous covariates. *Statistical Methods in Medical Research*, 32(12) :2331–2346, 2023.

Debashis Ghosh and Danyu Y Lin. Marginal regression models for recurrent and terminal events. *Statistica Sinica*, pages 663–688, 2002.

Yinan Huang, Jieni Li, Mai Li, and Rajender R Aparasu. Application of machine learning in predicting survival outcomes involving real-world data : a scoping review. *BMC Medical Research Methodology*, 23(1) :268, 2023.

Hemant Ishwaran, Udaya B Kogalur, Eugene H Blackstone, and Michael S Lauer. Random survival forests. 2008.

Jerald F Lawless and Claude Nadeau. Some simple robust methods for the analysis of recurrent events. *Technometrics*, 37(2) :158–168, 1995.

Virginie Rondeau, Yassin Marzroui, and Juan R Gonzalez. frailtypack : an r package for the analysis of correlated survival data with frailty models using penalized likelihood estimation or parametrical estimation. *Journal of Statistical Software*, 47 :1–28, 2012.