Random Survival Forests with An Example of Systolic Heart Failure

Frances Lin

March 2022

A Picture - Forests

Photo by Unilever

Background and Objective

Random forests (RF) has shown to be highly applicable and accurate, comparable to state-of-art methods such as bagging, boosting, and support vector machines. However, RF have primarily been used for classification and regression tasks.

On the other hand, methods that are commonly used for survival data rely on assumptions such as proportional hazards that is often too restrictive.

In this paper, Ishwaran et al. (2008) introduce random survival forests (RSF), an extension of Breiman's RF (2001). RSF incorporates survival information and is designed for analysis of right-censored data.

Outline

- Introduction: RF
- Introduction: RF vs RSF
- **Algorithm (High-Level)**
    - **Splitting Rules**
    - **Terminal Node Statistics: CHF (Cumulative Hazard Function) & Survival Function**
    - **OOB (out-of-bag) Prediction Error**
- Discussion
- A R Example (If Time Allows)

Introduction: RF

Random forests (RF) is made up from a collection of trees, and randomness is introduced in the following tree-growing process.

RF works by

1. first randomly drawing bootstrap sample of data and using it to grow a tree
2. and then at each node of the tree, randomly selecting subsets of predictors for splitting.

The final outcome is based on majority voting (for classification) or averaging (for regression).

RF is able to approximate rich classes of functions while keeping error low.

Introduction: RF vs RSF

RSF adheres strictly to Breiman's RF (2003). RF is designed such that the tree-growing process takes into account the outcome.

In right-censored survival settings, the splitting rule, the predicted results and the measure of prediction accuracy in RSF must incorporate survival information.

Algorithm (High-Level)

Random survival forests involves the following steps:

1. Randomly draw B bootstrap samples from the original data and grow a survival tree for each bootstrap sample.
2. At each node of the tree, randomly select p predictors.
3. Grow the tree to full size under the constraint that a terminal node shoud have no less than $d_0 > 0$ unique deaths.
4. Calculate CHF (cumulative hazard function) for each tree and average them to obtain the ensemble CHF.
5. Use OOB (out-of-bag) data to calculate prediction error for the ensemble CHF.

Note. OOB data is the number of samples that are NOT drawn, which makes up of $\sim$37% of the original data.

Splitting Rules

Terminal Node Statistics

The CHF and survival function for each terminal node h are estimated using the bootstrapped Nelson-Aalen and Kaplan-Meier estimators

$$H_h(t) = \sum_{t_{j,h} \leq t} \frac{d_{j,h}}{Y_{j,h}}, \qquad S_h(t) = \prod_{t_{j,h} \leq t} (1 - \frac{d_{j,h}}{Y_{j,h}}),$$

where $d_{j,h}$ is the number of deaths and $Y_{j,h}$ is the number of individuals at risk at time $t_{j,h}$.

Since the tree is binary, X will fall into a unique terminal node h, which is why the CHF and survival estimators equals to the Nelson-Aalen and Kaplan-Meier estimators for X's terminal node

$$H(t|X)^{IB} = H_h(t), \;\; S(t|X)^{IB} = S_h(t), \;\; X \in h.$$

Terminal Node Statistics

The ensemble CHF and survival function for IB (in-bag) and OOB (out-of-bag) are calculated by averaging the tree estimators

$$H(t|X)^{IB} = \frac{1}{ntree}\sum_{b=1}^{ntree} H_b(t|X), \quad S(t|X)^{IB} = \frac{1}{ntree}\sum_{b=1}^{ntree} S_b(t|X),$$

$$H_i^{OOB}(t) = \frac{1}{|O_i|}\sum_{b\in O_i} H_b^{IB}(t|X_i), \quad S_i^{OOB}(t) = \frac{1}{|O_i|}\sum_{b\in O_i} S_b^{IB}(t|X_i),$$

where O_i records the number of OOB case.

Note. IB estimators are used for prediction, whereas OOB estimators are used for inference (on the training data) and prediction error estimation.

Prediction Error: PE & C-Index

Prediction error (PE) is defined as $1 - C$, where C is Harrell's concordance index (C-index) (Harrell Jr et al., 1982). PE

- is between 0 and 1.
- measures how well the predictor correctly ranks two random individuals in terms of survival. (e.g. PE $= 0.5$: no better than random guessing.)

Harrell's C-index is a popular means for assessing prediction performance in survival settings. The C-index

- estimates the probability that, in a randomly selected pair of cases, the case that fails first had a *worst predicted outcome*.
- is interpreted as a misclassification probability.
- does not depend on choosing a fixed time for evaluation of the model.
- accounts for censoring of individuals.

Prediction Error: Mortality & Ensemble Mortality

To compute the C-index, we use mortality to define *worst predicted outcome*.

Discussion

A R Example (If Time Allows)

Data `peakVO2` can be loaded from **R** package `randomForestSRC`.

This survival data consists of $n = 2231$ adult patients with systolic heart failure (Hsich et al., 2011).

- All patients underwent cardiopulmonary stress testing, and a total of 742 patients died during a mean follow-up of 5 years (maximum for survivors, 11 years).
- The outcome is all-cause mortality, and a total of $p = 39$ predictors were measured for each patient including demographic, cardiac and noncardiac comorbidity, and stress testing information.

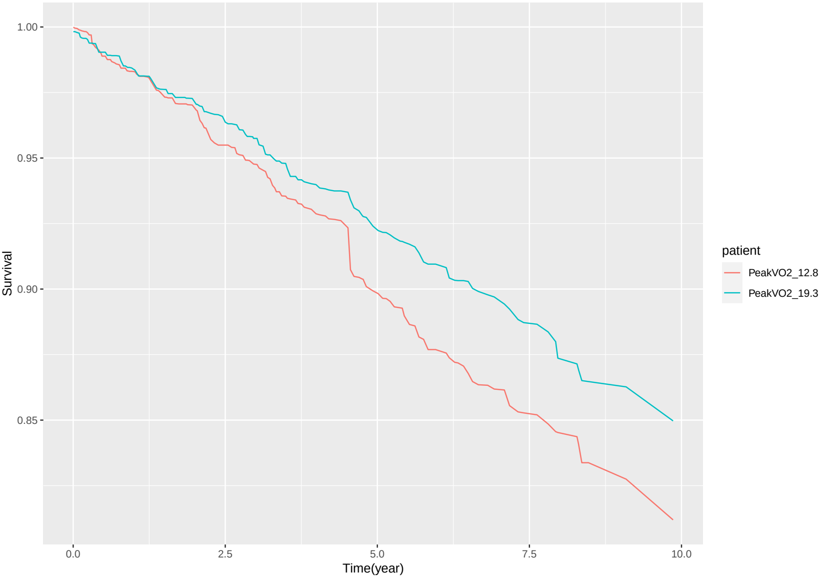
A R Example

RSF results using the `rfsrc` function of **R** package `randomForestSRC`:

```
                         Sample size: 2231
                    Number of deaths: 726
                     Number of trees: 1000
           Forest terminal node size: 5
       Average no. of terminal nodes: 259.547
No. of variables tried at each split: 7
              Total no. of variables: 39
       Resampling used to grow trees: swor
    Resample size used to grow trees: 1410
                            Analysis: RSF
                              Family: surv
                      Splitting rule: logrank *random*
       Number of random split points: 50
                          (OOB) CRPS: 0.15508379
   (OOB) Requested performance error: 0.29965793
```

A R Example

Plot of predicted survival curves of two hypothetical individuals:



Reference