PH 716 Applied Survival Analysis

Part 11: Random Survival Forest

Zhiyang Zhou (zhou67@uwm.edu, zhiyanggeezhou.github.io)

2026/05/07 15:56:50

Motivation

Cox PH models assume:

- relationships are relatively simple
- effects are linear
- variables behave consistently over time

Real clinical data are often much more complicated.

Random survival forest (RSF)

RSF is a machine learning method, extending decision-tree methods to survival data. Its classical version handles time-to-first-event data only, but there are also extensions for competing risks and recurrent events. It is especially useful when:

- there are many variables
- relationships are complex
- interactions are unknown
- prediction accuracy is important

How RSF works (conceptually)

Step 1: Create many bootstrap samples

To create a tree, randomly sample subjects from the training dataset with replacement. Each sample will produce one tree.

Step 2: Grow a survival tree

Given a subset of data, grow a tree by repeatedly dividing subjects into mutually exclusive groups with similar values of covariates. As the tree grows, groups become small.

Suppose we start with all subjects in the subset together. RSF asks questions such as:

- Is age > 60?
- Is tumor stage advanced?
- Is certain biomarker level high?

Each question divides the subjects into smaller groups. Over time, RSF creates many finalized groups (aka. terminal nodes) containing subjects with similar values of covariates.

Step 3: Estimate survival

For each terminal node, RSF estimates the survival curve via KM or NA methods.

Step 4: Prediction

- For a new subject with certain covariate values, in each tree, one may determine which terminal node the subject should belong to, following the same questions as in Step 2.
- Then a predicted survival probability is issued based on the survival curve of that terminal node.
- RSF combines all those predictions into one final prediction.

Ex. 11.1 Revisit `survival::pbc`

```r
options(digits=4)
library(survival)
library(randomForestSRC)
pbc = survival::pbc
pbc$status2 <- ifelse(pbc$status == 2, 1, 0)

# RSF
rsf_model <- rfsrc(
  Surv(time, status2) ~ age + bili + albumin + edema,
  data = pbc,
  ntree = 500,
  notesize = 15 # minimum number of subjects allowed in a terminal node
)
# Model summary
print(rsf_model)

# Variable importance
vimp_result <- vimp(rsf_model)
plot(vimp_result)

# Survival curves for subjects 1 to 3 in the training set
plot.survival(rsf_model, subset = c(1:3)) # What is OOB survival?

# Predict the survival probability for a new subject
new_data <- data.frame(age = 60, bili = 1.0, albumin = 3.5, edema = 0)
predicted_survival <- predict(rsf_model, newdata = new_data)

# Survival curves for a new subject
plot.survival(predicted_survival)
```

Out-of-bag (OOB) survival

- Recall that each tree is built using a bootstrap sample. To build one tree, RSF randomly samples 100 patients with replacement. There must be some subjects are left out. They are called OOB subjects.
- Each subject in the training set is OOB for many trees. Its OOB survival is the average of predicted survival from those trees.

Limitations of RSF

- Harder to interpret. Unlike simple regression models, RSF behaves more like a "black box''.
- Computationally intensive. Large forests can require substantial computing time.