1. Introduction

Random survival forest (RSF) is a random forest method used to analyze right deletion survival data. It introduces new survival splitting rules for growing survival trees and new missing data algorithms for estimating missing data.

RSF introduced the event retention principle for living forests and used it to define overall mortality, which is a simple interpretable mortality measure that can be used as a predictive result. R package "randomSurvivalForest" provides an interface to use.

2. RSF framework

(1) Extract B bootstrap samples from the original data, and each bootstrap sample excludes 37% of the data on average, which is called out-of-bag data (OOB data).
(2) Construct a binary survival tree for each bootstrap sample. At each node of the tree, p candidate variables are randomly selected, and the node is split by using the candidate variables that maximize the survival difference between the child nodes.
(3) Grow the tree to full size with at least $d_0 > 0$ number of events (deaths).
(4) Calculate cumulative risk function for each tree and obtain the mean value of the integrated cumulative risk function.
(5) Calculate the integrated cumulative risk function prediction error with OOB data.

3. Ensemble cumulative hazard

Regenerating the survival tree and building the integrated CHF (Cumulative Hazard Function) are the central elements of the RSF algorithm.

3.1 Binary Survival tree

Like CART, a survival tree is a binary tree generated by recursively splitting tree nodes. A tree grows from the root node, which is the top of the tree that contains all the data. Using predetermined survival criteria, the root node is divided into two children: left and right. In turn, each child node is split, producing left and right child nodes with each split. The process is repeated recursively for each subsequent node.

A good node segmentation can maximize the survival difference between the offspring. The optimal split for a node can be found by searching all possible x variables and split values c and selecting the x and c that maximize the survival difference. By maximizing survival differences, trees separate out different situations. Eventually, as the number of nodes increased, the alien cases were separated, and each node in the tree became homogenous, made up of samples with similar survival rates.

3.1.1 Node Segmentation Rules

At each node, a predictive variable x and a partition value c are randomly selected, c being some value of x. If $x_i < c$, then the sample i is divided into the right child node; If $x_i > c$, the sample i is assigned to the left child node.

Calculate log rank test:

$$L(x,c)=\frac{\sum_{i=1}^{N}(d_{i,1}-Y_{i,1}\frac{d_i}{Y_i})}{\sqrt{\sum_{i=1}^{N}\frac{Y_{i,1}}{Y_i}(1-\frac{Y_{i,1}}{Y_i})\frac{Y_i-d_i}{Y_i-1}d_i}}$$

Where, $j \in \{1,2\}$ represents left and right child nodes,
$d_{i,j}$ is the number of events occurring in the sub-node j at the time t_i,
$Y_{i,j}$ is the number of all patients in the sub-node j at the moment t_i,
d_i is the number of default events occurring at the time t_i, $d_i=\sum_j d_{i,j}$,
Y_i is the number of all borrowers at the moment t_i, $Y_i=\sum_j Y_{i,j}$.

Iterate over all possible variables x and partition values c, find variables x and partition values c that satisfies $|L(x^*,c^*)| \geq |L(x,c)|$ for all x^* and c^*.