

aorsf: An R package for supervised learning using the oblique random survival forest

Byron C. Jaeger¹, Sawyer Welden¹, Kristin Lenoir¹, and Nicholas M Pajewski¹

DOI:

¹ Wake Forest School of Medicine

Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Submitted:

Published:

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC-BY](#)).

Summary

The random forest (RF) is a supervised learning method that combines predictions from a large set (i.e., an ensemble) of decision trees (Breiman, 2001). To de-correlate decision trees in the ensemble, random subsets of training data are used to grow each tree and a subset of randomly selected predictor variables is considered at each non-terminal node in the trees. In Breiman (2001), both axis based and oblique RFs are described. Axis-based trees split non-terminal nodes using individual predictor variables whereas oblique trees use a linear combination of variables. Although oblique RFs outperform their axis-based counterparts in prediction tasks (Menze, Kelm, Splitthoff, Koethe, & Hamprecht, 2011), they are also more computationally intensive and less easily interpreted. Thus, software packages have focused mostly on axis based RFs, with few packages supporting oblique RFs, and even fewer supporting oblique random survival forests (ORSFs; Jaeger et al. (2019)).

aorsf is an R package optimized for fitting, interpreting, and computing predictions with ORSFs. The target audience includes both **practitioners** aiming to develop an accurate and interpretable risk prediction model (e.g., see Segar et al. (2021)) and **researchers** who want to conduct experiments comparing different techniques for identifying linear combinations of predictor variables in a controlled environment (e.g., see Katuwal, Suganthan, & Zhang (2020)). Key features of **aorsf** include computational efficiency compared to existing software, extensive unit and integration testing that ensure cross-platform consistency and reproducibility, and user-friendly documentation paired with an application programming interface that facilitates proper usage of the core algorithms. Interpretation is facilitated by partial dependence and negation importance, a novel method for variable importance designed for compatibility with oblique decision trees.

Existing software

The **obliqueRF** R package supports classification and regression using oblique random forests but does not support survival analysis. The **ranger** and **randomForestSRC** packages support survival analysis but not oblique decision trees. The **obliqueRSF** R package is currently the only software available to fit ORSFs. Penalized Cox regression models are used to identify linear combinations of inputs. **aorsf** is a re-write of the **obliqueRSF** package, with a focus on computational efficiency and flexibility. For example, **aorsf** allows oblique decision trees to be grown using Newton Raphson scoring (the default), penalized Cox regression, or a user-defined function.

Newton Raphson scoring

The default routine for creating linear combinations of predictor variables in **aorsf** applies Newton Raphson scoring to the partial likelihood function of the Cox regression model. **aorsf** uses the same approach as the **survival** package to complete this estimation procedure efficiently. Full details on the steps involved have been made available by Therneau (2022). Briefly, a vector of estimated regression coefficients, $\hat{\beta}$, is updated in each step of the procedure based on its first derivative, $U(\hat{\beta})$, and second derivative, $H(\hat{\beta})$:

$$\hat{\beta}^{k+1} = \hat{\beta}^k + U(\hat{\beta} = \hat{\beta}^k) H^{-1}(\hat{\beta} = \hat{\beta}^k)$$

While it is standard practice in statistical modeling to iterate until a convergence threshold is met, the default approach in **aorsf** only completes one iteration. Our decision to implement this design is based on three points. First, while completing more iterations reduces bias in the regression coefficients, it generally amounts to little or no gain in prediction accuracy for the random forest due to the bias-variance trade-off. Second, computing U and H requires computation and exponentiation of the vector $X\hat{\beta}$, where X is the matrix of predictor values, but these steps can be skipped on the first iteration if an initial value of $\hat{\beta} = 0$ is assumed, allowing for a reduction in required computation. Third, using only one iteration with a starting value of 0 for $\hat{\beta}$ ensures numerical stability by avoiding exponentiation of large numbers, which can occur in later iterations depending on the scale of variables in X .

Computational efficiency

The increased efficiency of **aorsf** versus **obliqueRSF** is explained by improved memory management use of fast Newton Raphson scoring instead of penalized Cox regression (the default approach in **obliqueRSF**). The benchmark below shows **aorsf** is about 2.5 times faster than **obliqueRSF** when both packages use penalized regression and about 400 times faster when **aorsf** uses Newton Raphson scoring, suggesting that the increased efficiency of **aorsf** is largely attributable to its use of Newton Raphson scoring.

```
library(microbenchmark)
library(obliqueRSF)
library(aorsf)

options(microbenchmark.unit="relative")

data_bench <- pbc_orsf
data_bench$id <- NULL

microbenchmark(

  obliqueRSF = ORSF(data = data_bench,
                    verbose = FALSE,
                    ntree = 100),

  aorsf_net = orsf(data_train = data_bench,
                  formula = time + status ~ .,
                  control = orsf_control_net(),
                  n_tree = 100),

  aorsf = orsf(data_train = data_bench,
               formula = time + status ~ .,
```

```
n_tree = 100),

times = 10

)
```

Unit: relative

##	expr	min	lq	mean	median	uq	max	neval	cld
##	obliqueRSF	378.1932	390.2034	415.7029	415.6862	433.6971	494.8694	10	c
##	aorsf_net	72.4856	118.3790	128.7465	137.5481	139.2145	148.4609	10	b
##	aorsf	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	10	a

Acknowledgements

The development of this software was supported by the Center for Biomedical Informatics at Wake Forest School of Medicine.

References

- Breiman, L. (2001). Random forests. *Machine learning*, 45(1), 5–32.
- Jaeger, B. C., Long, D. L., Long, D. M., Sims, M., Szychowski, J. M., Min, Y.-I., McClure, L. A., et al. (2019). Oblique random survival forests. *The Annals of Applied Statistics*, 13(3), 1847–1883.
- Katuwal, R., Suganthan, P. N., & Zhang, L. (2020). Heterogeneous oblique random forest. *Pattern Recognition*, 99, 107078.
- Menze, B. H., Kelm, B. M., Splitthoff, D. N., Koethe, U., & Hamprecht, F. A. (2011). On oblique random forests. *Joint european conference on machine learning and knowledge discovery in databases* (pp. 453–469). Springer.
- Segar, M. W., Jaeger, B. C., Patel, K. V., Nambi, V., Ndumele, C. E., Correa, A., Butler, J., et al. (2021). Development and validation of machine learning–based race-specific models to predict 10-year risk of heart failure: A multicohort analysis. *Circulation*, 143(24), 2370–2383.
- Therneau, T. (2022, April). Survival package source code documentation. Retrieved from <https://github.com/therneau/survival/blob/5440691d44abea537b08aeb60153a31654d66a9b/noweb>