A Report on "Censoring Unbiased Regression Trees and Ensembles"

SEOKHO LIM, University of Waterloo, Canada

1 INTRODUCTION

Steingrimsson et al. (2019) extended the theory of censoring unbiased transformations (CUTs) and developed two new classes of algorithms for constructing surival trees and survival forests: censoring unbiased regression trees (CURTs) and censoring unbiased regression ensembles (CUREs). Their generalization of a CUT from Rubin and van der Laan (2007) is proven to be a CUT by Theorem 3.1 in the article, and in the case of using the squared loss function, Theorem 4.1 shows that after imputing a censored dataset, it is possible to implement the squared loss variations CURT$-L_2$ and CURE$-L_2$ using the existing software packages.

In this report, a survey of the development of CURTs and CUREs is provided, closely following the manner of the original article. Then, the Weibull model, the default random forest, the doubly robust CUT model with L_2 loss, and the Buckley-James CUT model with L_2 loss are implemented. Note that the default random forest is a default ensemble, while the doubly robust CUT model with L_2 loss and the Buckley-James CUT model with L_2 loss are CURE$-L_2$. Finally, using the datasets from the TRACE Study and the Copenhagen Stroke Study, the four models will predict survival probabilities after 3 years and the performance is compared using the Brier Score.

1.1 A Preliminary to the Many Formulas To Come

Throughout this report, the following definitions from the original article are used: $(Z, W')'$ represents a vector of data, with $Z \in \mathbb{R}$ and $W \in S \subseteq \mathbb{R}^p$ where $W = (W^{(1)}, ..., W^{(p)})$ which is continuous or ordinal; $\psi : S \to \mathbb{R}$ is a real-valued function of W; $L(Z, \psi(W))$ is a general loss function; and $L_2(Z, \psi(W)) = (Z - \psi(W))^2$ is the squared loss function.

2 CLASSIFICATION AND REGRESSION TREES

In the original article, the authors give a brief introduction to CART, originally developed by Breiman (1984). Algorithm 1 in the article is a general implementation of CART, and Algorithm 2 in the article is a general implementation of CURE.

In this report, a modified version of each algorithm is provided. For Algorithm 1, it is the exact algorithm that was used to fit a regression tree. Algorithm 2 is not used, because as noted later, there is no existing software package that implements the required general weight bootstrapping. Instead, the fitted trees from Algorithm 1 are averaged to create an ensemble. Implementations are provided in R code on Crowdmark.

Author's address: Seokho Lim, s32lim@uwaterloo.ca, University of Waterloo, Waterloo, Ontario, Canada.

After determining a set of desired stopping conditions C, which in this report is simply checking if there is a child node that can be further split, we run the following algorithm to construct a regression tree.

Algorithm 1 CART Algorithm (modified)

```
Let the current node be the root node, containing all data in $\mathcal{F}$.
while $C$ not met do
    In the current node, identify all possible binary splits for covariates $W^{(j)}$ for $j = 1, ..., p$, in the form of $W^{(j)} \leq c$ or $W^{(j)} > c$.
    Choose the split with the largest reduction in $\sum_{i \in node} L(Z_i, \phi(W_i))$, and divide it into two mutually exclusive new nodes.
    Let a remaining child node to be the current node.
end while
```

Algorithm 2 in the original paper describes the algorithm to build a general ensemble using custom weights for bootstrapping. The authors noted that they were not able to find in the existing literature an available software package to implement the general weight bootstrapping.

Algorithm 2 CURE Algorithm (modified)

```
Generate M independent sets of exchangeable bootstrap weights $\omega_{1m}, ..., \omega_{nm}$ where $m = 1, ..., M$.
for $m = 1, ..., M$ do
    Build a CART tree $\psi_m$ using Algorithm 1, replacing the original loss function with $\sum_{i \in node} \omega_{im} L(Z_i, \phi(W_i))$.
end for
Calculate an estimator for terminal nodes in each $\psi_m$, and average over to get the final ensemble predictor.
```

Therefore, in the context of this report, trees constructed using Algorithm 1 used nonparametric bootstrapping, and then an ensemble was created by averaging with equal weights.

3 CENSORING UNBIASED TRANSFORMATIONS

Central to deriving the new class of algorithms is the theory of CUTs. Following the authors' notation, let Y be a scalar function of $(Z, W')'$, which denotes the full data, and let $Y^*(O)$ be a function of $O = (\tilde{Z}, \Delta, W')'$. We define $Y^*(O)$ to be a CUT for Y if the following condition holds for every $\omega \in \mathcal{S}$:

$$E[Y^*(O)|W = \omega] = E[Y|W = \omega] \tag{1}$$

One of the earliest known example of a CUT is the following:

$$Y^*(O) = \Delta T + (1 - \Delta)E[T|T > t, W = \omega] \tag{2}$$

3.1 Generalization of a CUT

To introduce the authors' generalization of the CUT from Rubin and van der Laan (2007), we define the following: let $\psi(r,\omega), (r,\omega) \in R^* \times \mathcal{S}$ be a known continuous scalar function for $r \in R^*$ except possibly at a finite number of points, and let $G(t|\omega)$ and $S(t|\omega)$ be two functions on $R^+ \times \mathcal{S}$. Also, define $\Lambda_G(t|\omega) = -\int_0^t [G(u|\omega)]^{-1} dG(u|\omega)$ and $m_\phi(t,\omega;S) = [S(t|\omega)]^{-1} \int_t^\infty \phi(h(u),\omega) dF(u|\omega)$.

Then, under certain conditions, the authors introduced the following transformation, which was proven by Theorem 3.1 in the article to be a CUT:

$$Y_d^*(O;G,S) = \frac{\Delta\phi(\tilde{Z},W)}{G(\tilde{T}|W)} + \frac{1-\Delta}{G(\tilde{T}|W)} m_\phi(\tilde{T},W;S) - \int_0^{\tilde{T}} \frac{m_\phi(u,W;S)}{G(u|W} d\Lambda_G(u|W) \tag{3}$$

An alternative CUT is obtained by supposing $G(t|\omega) = 1$ for all $(t,\omega) \in R^+ \times \mathcal{S}$, which is named the Buckley-James CUT:

$$Y_b^*(O;S) = \Delta\phi(\tilde{Z},W) + (1-\Delta) m_\phi(\tilde{T},W;S) \tag{4}$$

Note that this CUT is a generalization of (2), which is obtained by setting $\phi(h(u),\omega) = u$.

3.2 Loss Functions from CUTs

The authors derived loss functions that correspond to the two CUTs: the doubly robust loss function; and the Buckley-James loss function. The doubly robust loss function is the following:

$$L_d(O;\psi;G,S) = \frac{\Delta L(\tilde{Z},\psi(W))}{G(\tilde{T}|W)} + \frac{1-\Delta}{G(\tilde{T}|W)} m_L(\tilde{T},W;S) - \int_0^{\tilde{T}} \frac{m_L(u,W;S)}{G(u|W} d\Lambda_G(u|W) \tag{5}$$

As an alternative, the Buckley-James Loss function is provided:

$$L_b(O,\psi;S) = \Delta L(\tilde{Z},\psi(W)) + (1-\Delta) m_L(\tilde{T},W;S) \tag{6}$$

For applications to the TRACE Study and the Copenhagen Stroke Study, both L_d and L_b are used respectively to implement the doubly robust CUT model and the Buckley-James CUT model.

3.3 A Special Loss Function - L_2

A condition in which we can easily implement CURT and CURE in existing software packages is that we use the squared loss function in L_d and L_b. By replacing $L(\tilde{Z},\psi(W))$ with $L_2(\tilde{Z},\psi(W))$ in L_d and L_b, the authors derived an equivalent form of each corresponding loss function, and a way to impute any dataset, which are omitted from this report due to their complexity. However, a brief introduction is given in the following section.

4 IMPLEMENTATION OF CURT-L_2 AND CURE-L_2

The authors derived an equivalent representation for $L_{2,d}(O_i,\psi;G,S)$ by defining the following A_{ki}, B_{ki}, C_{ki} for $k = 0, 1, 2$:

$$A_{ki}(G) = \frac{\Delta_i \tilde{Z}_i^k}{G(\tilde{T}_i|W_i)} \tag{7}$$

$$B_{ki}(G,S) = \frac{(1-\Delta_i)m_k(\tilde{T}_i, W_i; S)}{G(\tilde{T}_i|W_i)} \tag{8}$$

$$C_{ki}(G,S) = \int_0^{\tilde{T}_i} \frac{m_k(u, W_i; S)d\Lambda_G(u|W_i)}{G(u|W_i)} \tag{9}$$

Then, their Theorem 4.1 showed that one can simply impute the censored dataset and then use the existing software packages to implement the doubly robust CUT model with L_2 loss and the Buckley-James CUT model with L_2 loss.

5 APPLICATIONS

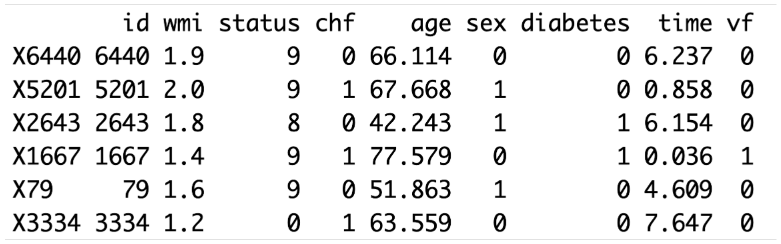
After imputing a censored dataset, the available R packages such as `randomForest` and `randomForestSRC` provide an easy way to implement the doubly robust CUT model with L_2 loss and the Buckley-James CUT model with L_2 loss. One of the authors, Dr. Steingrimsson, provided a sample code to work from. To make it possible to replicate the result of this study, random seed was set at 437.

Parameter tuning is an important part of fitting trees and forests. In the original article, the authors had 1000 trees, with `mtry`, the number of covariates to be considered for splitting, as $\sqrt{p}$ where p denotes the number of covariates in train set. For other parameters such as `nodesize`, default values were used.

There are four models to be implemented: the Weibull model from STAT 437; the default random forest (default RF); the doubly robust CUT model with L_2 loss (DR L2); and the Buckley-James CUT model with L_2 loss (BJ L2). Then, brier score will be used to compare the performanace of predicting a survival probabilty after 3 years.

5.1 The TRACE Study

The TRAndolapril Cardiac Evaluation, abbreviated as the TRACE study, evaluates the effect of trandolapril on angiotensin-converting enzyme inhibition, and investigates the mortality of the patients after experiencing myocardial infarction. In the context of the original article, the event of interest is death from the myocardial infraction. The authors considered a subset of the patients who survived past 30 days, after removing two patients with an undefined censoring status; the final dataset consists of 1689 patients. Figure 1 shows the first glance at the TRACE Study dataset.



```
         id wmi status chf    age sex diabetes  time vf
X6440 6440 1.9      9   0 66.114   0        0 6.237  0
X5201 5201 2.0      9   1 67.668   1        0 0.858  0
X2643 2643 1.8      8   0 42.243   1        1 6.154  0
X1667 1667 1.4      9   1 77.579   0        1 0.036  1
X79     79 1.6      9   0 51.863   1        0 4.609  0
X3334 3334 1.2      0   1 63.559   0        0 7.647  0
```

Fig. 1. The first 6 rows of the TRACE Study dataset

The four models were implemented and used to predict the survival probabilty after 3 years. For the Weibull model, the model was implemented using `survreg` from `survival` package, with no dataset imputation. For the default RF, DR L2, and BJ L2, 1000 trees were averaged to predict the terminal nodes for each fold in 10-fold cross validation, and then ten brier scores from the cross validation were averaged to produce the values in Table 1.

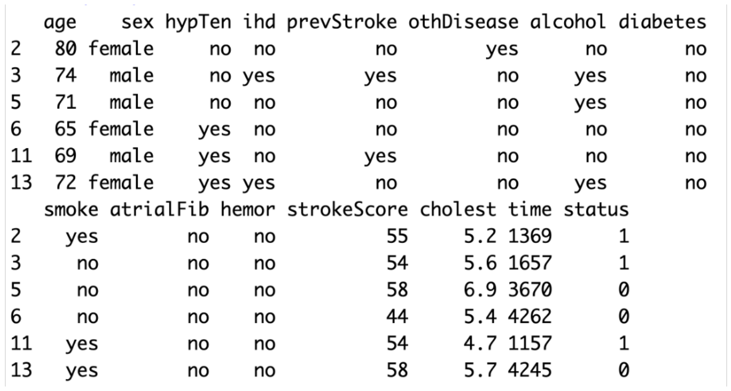
Table 1. Averaged Brier Score for the TRACE Study

Weibull	Default RF	DR L2	BJ L2
0.1643	0.1681	0.1665	0.1661

Interestingly, the performance for the default RF, DR L2, and BJ L2 was slightly better than the average results shown in the article. Moreover, the Weibull model performed the best by achieving the lowest averaged Brier Score for the TRACE dataset.

5.2 The Copenhagen Stroke Study

The Copenhagen Stroke Study consists of 1,197 patients with acute stroke who underwent acute stroke care. The study investigates the effect of organized stroke care and the recovery owing to several factors, such as whether or not a patient smokes, or whether or not a patient is diabetic. The dataset from `pec` consists of 518 patients, and in the context of the original article, the event of interest is time from admission to a caring unit to death. Figure 2 shows the first glance at the Stroke Study dataset.



```
   age    sex hypTen ihd prevStroke othDisease alcohol diabetes
2   80 female     no  no         no        yes      no       no
3   74   male     no yes        yes         no     yes       no
5   71   male     no  no         no         no     yes       no
6   65 female    yes  no         no         no      no       no
11  69   male    yes  no        yes         no      no       no
13  72 female    yes yes         no         no     yes       no
   smoke atrialFib hemor strokeScore cholest time status
2    yes        no    no          55     5.2 1369      1
3     no        no    no          54     5.6 1657      1
5     no        no    no          58     6.9 3670      0
6     no        no    no          44     5.4 4262      0
11   yes        no    no          54     4.7 1157      1
13   yes        no    no          58     5.7 4245      0
```

Fig. 2. The first 6 rows of the Copenhagen Stroke Study dataset

Then, the four models were implemented in the same form as in the TRACE Study application to produce the following table.

Table 2. Averaged Brier Score for the Copenhagen Stroke Study

Weibull	Default RF	DR L2	BJ L2
0.1951	0.1968	0.1958	0.1957

The performance for the default RF, DR L2, and BJ L2 was comparable to the results for each model in the article. Interestingly, the Weibull model again performed the best by achieving the lowest averaged Brier Score for the Copenhagen Stroke Study dataset.

6 CONCLUSION

While some of the key findings of this report, including understanding of the original article and implementation of the algorithms, are dispersed throughout the report and the submitted R code, here in this section a couple of reflection points is provided.

It is difficult to directly compare the Weibull model to the other three models, despite the fact that the Weibull model achieves the best average Brier Score. For the default RF, DR L2, and BJ L2, there are tuning parameters to work with, which can improve the performance of the models. Also, they are different types of models; the Weibull model is an example of a parametrized model, while the other three models are non-parametrized.

However, the complexity of the underlying theories behind DR L2 and BJ L2, when compared to a simple implementation of the Weibull model introduced in STAT 437, may be hard to justify, considering that their Brier Scores were slightly worse than the Weibull model's Brier Score, in both of the TRACE Study dataset and the Copenhagen Stroke Study dataset.

Still, the default RF, DR L2, and BJ L2 provide an easy way to interpret the models, as they consist of trees in which the data are split according to the best reduction in the loss functions. Therefore, it is more intuitive to interpret how a certain subset of the dataset falls into a terminal node.

7 BIBLIOGRAPHY

Breiman, L., Friedman, J. H., Stone, C. J., and Olshen, R. A. (1984), Classification and Regression Trees, New York: Chapman and Hall/CRC

Rubin, D., and van der Laan, M. J. (2007), "A Doubly Robust Censoring Unbiased Transformation," The International Journal of Biostatistics, 3, 1–21.

Steingrimsson, J.A., Diao, L., Molinaro, A.M. and Strawderman, R.L. (2019), "Censoring Unbiased Regression Trees and Ensembles," Journal of the American Statistical Association, 114(525), 370-383.

A APPENDIX

A.1 Revealing the Structure of a Fitted Tree

It is easy to take a glance at the structure of a fitted tree. From `randomForest` package, run `getTree` to get the following result.

```
   left daughter right daughter split var split point status prediction
1              2              3       age     61.6655     -3  1.3263302
2              4              5       age     60.9390     -3  2.1171196
3              6              7       age     62.3535     -3  0.9550031
4              8              9       age     54.6155     -3  2.0257791
5             10             11       age     60.9610     -3  3.7331431
6             12             13       age     62.2235     -3 -0.4882919
7             14             15       chf      0.5000     -3  1.0070137
8             16             17        vf      0.5000     -3  2.4795735
9             18             19       chf      0.5000     -3  1.5759137
10             0              0      <NA>      0.0000     -1  8.8898121
```

Fig. 3. The first 10 rows of `getTree` for a fitted tree from DR L2

The top node of the fitted tree corresponds to the first row of Figure 3; and the values in `left daughter` and `right daughter` columns correspond to the index of the table. By setting the parameter `labelVar = TRUE`, one can see which covariate was used to split the node (under `split var` column) and `split point` column denotes the value of splitting.