The Influence of Loss Functions on Oblique Survival Tree Induction

Malgorzata Kretowska
Bialystok University of Technology
Bialystok, Poland *m.kretowska@pb.edu.pl*

Marek Kretowski
Bialystok University of Technology
Bialystok, Poland *m.kretowski@pb.edu.pl*

Abstract

Survival trees are a common machine learning tool designed to handle censored data, where only partial information about failure events is available. Most survival tree models work by recursively dividing the feature space using splits defined by single attributes in internal nodes. However, there is a less common type known as oblique survival trees, which use more attributes to create splits in the form of hyperplanes. In this paper, we depart from the typical top-down approach and focus on globally induced oblique survival trees, aiming to optimize both prediction accuracy and model complexity. We propose using two different loss functions— the integrated Brier score and a likelihood-based loss— in the process of oblique survival tree induction. We then compare the resulting models in terms of their predictive performance and complexity.

Keywords: survival tree, evolutionary algorithm, integrated Brier score, Harrell's C-index

1. Introduction

Survival analysis comprises methods designed for a specific type of data, where the time until a predetermined event occurs is studied. While it may resemble a regression problem, there are unique considerations when analyzing this type of data. A common scenario involves what are known as censored observations, where the exact time of the event (often called a failure) is unknown. There are various types of censoring, with right censoring being the most prevalent.

In this context, given an observation (like a patient), time is measured from a defined starting event t_0 (such as disease diagnosis or surgery) to the earliest of the following (t_{end}): i) the occurrence of the event (failure), ii) the end of the observation period, or iii) loss to follow-up. The first option provides the precise failure time (uncensored observations), while the latter two scenarios indicate that the object has not experienced the event of interest up to that point (censored cases). Therefore, for uncensored individuals, $t = t_{end} - t_0$ represents the failure time, while for censored cases, it indicates the follow-up time. Survival data typically includes two types of observations: uncensored and censored.

When creating tools to analyze survival data, it is crucial to consider the incomplete information found in censored cases. Ignoring this can result in biased models. The most common statistical models for survival analysis include nonparametric methods like the Kaplan-Meier survival function [16] and the Nelson-Aalen cumulative hazard function [24], [1], as well as semi-parametric models such as Cox's proportional hazards model [5]. Additionally, there are parametric models like the Weibull, log-logistic, or log-normal distribution models [15]. However, statistical models often rely on certain assumptions, which may not always hold true. As an alternative, various machine learning tools, including tree-based models, have been developed over the years to address this issue.

The induction process for most proposed tree models follows a top-down approach. To achieve a final model with satisfactory generalization ability, two phases are required: induction and pruning. Induction aims to minimize a specified impurity measure [6],[28] or employs a between-node separation measure, often chosen from the Tarone–Ware class of two-sample statistics for censored data [4], [26]. The subsequent pruning phase utilizes cost-complexity pruning [3], and for survival trees, split-complexity pruning [23].

Another approach, known as the conditional inference framework, was introduced by Hothorn et al. [11]. This method makes decisions on split importance during node creation, eliminating the need for additional pruning steps. The splitting process halts when the test result in a given node is not statistically significant at a specified value of α. This idea was also utilized in the solution proposed by Kundu and Ghosh [20]. A non-greedy method of survival tree induction was introduced in [2], where the optimized function considers both the performance error of the tree on the training set and its complexity. This algorithm employs mixed-integer optimization and local search techniques to generate the resulting tree. Wang et al. [29] and Wang and Zhou [30] proposed using random forests to analyze survival data. Additionally, Wang et al. [31] provides a survey of machine learning methods applied to survival analysis.

Oblique trees, where tests in internal nodes are in the form of a hyperplane instead of a single variable, are less common solutions. Kretowska [17] proposed a dipolar survival tree induced by using convex, piece-wise linear criterion functions based on the idea of a dipole - a pair of observations. Recently, an oblique random survival forest was introduced by Jaeger et al. [13, 14].

The presence of censored observations necessitates evaluation measures that consider the partial information available. Among the most widely used metrics is the integrated Brier score (IBS) [9], which examines the disparity between the actual and predicted survival functions for a given observation. Another popular metric is Harrell's C-index [10], which assesses the concordance between the actual and predicted times of failure occurrence.

In this paper, we apply the global induction of oblique survival trees introduced in [18] to investigate the influence of two types of loss functions: the integrated Brier score and a likelihood loss incorporated into a fitness function used in an evolutionary algorithm. Each of these criteria considers the presence of censored observations, emphasizing certain aspects of the models while overlooking others. By considering different loss functions, we aim to compare their effects on the induced tree structures, focusing on both complexity and generalization ability, assessed by two metrics: the integrated Brier score and Harrell's C-index. The experiments were conducted using five medical datasets.

The paper consists of six sections. Section 2 introduces basic information about survival analysis, including data representation, distribution functions, and oblique survival trees. Section 3 describes performance measures and loss functions used in evolutionary induction, as well as in the evaluation of the obtained survival models. In Section 4, we introduce the basic concepts of evolutionary induction of survival trees, while Section 5 presents the obtained results. Section 6 summarizes the results.

2. Survival Analysis

We assume having a learning set, L, which consists of M observations. In survival analysis, the ith observation is described by a set of three values $(\mathbf{x}_i, t_i, \delta_i)$, where $\mathbf{x}_i$ is the N-dimensional feature vector, t_i is the observed time, which for uncensored subjects is equal to its failure time, for censored - it takes values of censoring time, δ_i is the failure indicator, which takes one of two values: 0 for censored observations or 1 otherwise.

2.1. Distributions of Survival Time

The distribution of the survival time may be represented by several functions. One of them is a survival function $S(t) = P(T > t)$, which gives the probability of surviving beyond the time t.

The other, commonly used function is a hazard function, calculated as:

$$\lambda(t) = \lim_{\Delta t \to 0} \frac{P(t \leq T < t + \Delta t | T \geq t}{\Delta t}, \tag{1}$$

where $\lambda(t)\Delta$ is the probability of failure in the in infinitesimal interval $(t, t+\Delta t)$, given survival at time t. A cumulative hazard function, $\Lambda(t)$, is directly associated with $\lambda(t)$:

$$\Lambda(t) = \int_0^t \lambda(s)ds \tag{2}$$

and can be interpret as the cumulative amount of hazard up to time t and is linked with the survival function: $\Lambda(t) = -\log S(t)$, so $S(t) = e^{-\Lambda(t)}$.

One of the most common estimators of the survival function is the Kaplan–Meier (KM) method [16]. If we assume that the events of interest occur at D distinct times $t_{(1)} < t_{(2)} < \ldots < t_{(D)}$, it is calculated as follows:

$$\hat{S}(t) = \hat{S}_{KM}(t) = \prod_{j|t_{(j)} \leq t} \left(\frac{m_j - d_j}{m_j} \right), \tag{3}$$

where d_j is the number of events at time $t_{(j)}$ and m_j is the number of patients at risk at $t_{(j)}$ (i.e., the number of patients who are alive at $t_{(j)}$ or experience the event of interest at $t_{(j)}$). The Nelson–Aalen estimator [24], [1] of the cumulative hazard function is qual to:

$$\hat{\Lambda}(t) = \sum_{j|t_{(j)} \leq t} \left(\frac{d_j}{m_j} \right) \tag{4}$$

and hence, the Nelson–Aalen estimator of survival function is calculated as:

$$\hat{S}_{NA}(t) = e^{-\hat{\Lambda}(t)} = \exp\left(\sum_{j|t_{(j)} \leq t} \frac{d_j}{m_j} \right). \tag{5}$$

2.2. Oblique Survival Tree

The primary goal of survival trees is to divide the N-dimensional feature space recursively to create large regions with similar survival patterns. The tree consists of internal and terminal nodes. Each internal node holds a split, which in univariate trees depends on a single feature (e.g., $x_i = c$). In oblique trees, this split takes the form of any hyperplane $H(w, c) = \{x : w^T x = c\}$, where many features are considered.

In traditional decision trees, terminal nodes are assigned a class number to which observations reaching the node belong. However, for survival trees, terminal nodes typically represent not just a single value, but a function. In our proposed method, each terminal node is defined by the Kaplan–Meier survival function, along with the calculated median survival time (Figure 1).

3. Performance Measures and Loss Function

In survival analysis, traditional performance measures such as mean square error or mean absolute error cannot be used due to censored observations, where the exact time of failure occurrence is unknown. The proposed indices are designed to utilize the available information from censored cases.

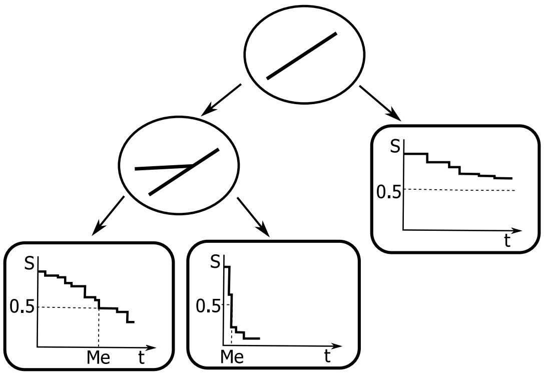

Fig. 1. Oblique survival tree.

3.1. Harrell's C-index

One of the widely used performance measures in survival analysis is Harrell's C-index [10], which is based on the numbers of concordant, discordant, and tied pairs of observations among all comparable pairs. A pair $(\mathbf{x}_i, \mathbf{x}_j)$ is considered comparable if we can order failure times of both observations. Therefore, we exclude from this set all pairs where: i) $\delta_i = \delta_j = 0$ or ii) $\delta_i = 0$ and $\delta_j = 1$ with $t_i < t_j$.

For a comparable pair of observations, if the predicted order of failure time occurrence, $\hat{t}$, matches the actual order, we call this pair concordant. Conversely, if the predicted order differs, the pair is discordant. Tied pairs occur when the predicted survival times for two observations are equal.

Hence, we can calculate the number of pairs of these three types as follows: the number of concordant pairs: $\#Conc = \sum_{i,j} \mathbb{1}(t_i > t_j)\mathbb{1}(\hat{t}_i > \hat{t}_j)\delta_j$, the number of discordant pairs: $\#Disc = \sum_{i,j} \mathbb{1}(t_i > t_j)\mathbb{1}(\hat{t}_i < \hat{t}_j)\delta_j$, and the number of tied pairs: $\#Tied = \sum_{i,j} \mathbb{1}(t_i > t_j)\mathbb{1}(\hat{t}_i = \hat{t}_j)\delta_j$ where $\hat{t}_i$ means the predicted survival time for the ith observation.

Harrell's C-index is calculated using the formula:

$$HCI = \frac{\#Conc + 0.5\#Tied}{\#Conc + \#Disc + \#Tied}. \tag{6}$$

3.2. Integrated Brier Score

The Integrated Brier Score (IBS) measures the difference between survival functions [9]. For a fixed time, t, we can look at three types of contributions for a given observation $\mathbf{x}_i$:

1. if $t_i \leq t$ and $\delta_i = 1$: the failure occurred before t, and the event status at t is equal to 0, so the contribution to the Brier score is $(0 - \hat{S}(t|\mathbf{x}_i))^2 = \hat{S}(t|\mathbf{x}_i)^2$;

2. if $t_i > t$ and ($\delta_i = 1$ or $\delta_i = 0$): the observations do not experience any event at time t; hence, the event status at t is equal to 1, and the contribution to the Brier score is $(1 - \hat{S}(t|\mathbf{x}_i))^2$;

3. if $t_i \leq t$ and $\delta_i = 0$: the contribution to the Brier score cannot be calculated because the event status at t is unknown for the observations.

Since the observations from group 3 do not contribute to the Brier score, we need to adjust by giving weights to the existing contributions. Observations from group 1 have a weight of $\hat{G}(t_i)^{-1}$, while those from group 2 have a weight of $\hat{G}(t)^{-1}$, where $\hat{G}(t)$ is the Kaplan–Meier estimator of the censoring distribution. It's calculated based on observations $(t_i, 1 - \delta_i)$. The

Brier score is defined as:

$$\begin{aligned} BS(t) = \ & \tfrac{1}{n}\textstyle\sum_{i=1}^{N}(\hat{S}(t|\mathbf{x}_i)^2 I(t_i \leq t \wedge \delta_i = 1)\hat{G}(t_i)^{-1} \\ & +(1-\hat{S}(t|\mathbf{x}_i))^2 I(t_i > t)\hat{G}(t)^{-1}), \end{aligned} \tag{7}$$

where $I(condition)$ is equal to 1 if the condition is fulfilled and zero otherwise. The IBS is calculated as follows:

$$IBS = \frac{1}{\max_{t_i}} \int_0^{\max_{t_i}} BS(t)dt. \tag{8}$$

3.3. Likelihood-Based Loss

A loss function proposed by LeBlanc and Crowley in [22] is derived from Cox's proportional hazard model [5], where the hazard function is defined as: $\lambda(t|\mathbf{x}) = \lambda_0(t)exp(\beta^T\mathbf{x})$. Here, $\lambda_0(t)$ is the baseline hazard and β is an N-dimensional vector of parameters. With a survival tree T having terminal nodes $h \in T$, the full likelihood of the learning set can be expressed as:

$$L = \prod_{h\in T}\prod_{\mathbf{X}_i\in h}(\lambda_h(t_i))^{\delta_i}e^{-\Lambda_h(t_i)}, \tag{9}$$

where λ_h and Λ_h represent the hazard and cumulative hazard functions for node h. Assuming $\lambda_h(t) = \theta_h\lambda_0(t)$, we get:

$$L = \prod_{h\in T}\prod_{\mathbf{X}_i\in h}(\theta_h\lambda_0(t_i))^{\delta_i}e^{-\theta_h\Lambda_0(t_i)}, \tag{10}$$

where $\Lambda_0(t)$ is the baseline cumulative hazard function and is calculated using the Nelson-Aalen estimator (4). The maximum likelihood estimator of θ_h, for $h \in T$ is:

$$\hat{\theta}_h = \frac{\sum_{\mathbf{X}_i\in h}\delta_i}{\sum_{\mathbf{X}_i\in h}\hat{\Lambda}_0(t_i)}. \tag{11}$$

Assuming a saturated model with only one observation in each terminal node, we find $\hat{\theta}_i^{sat} = \frac{\delta_i}{\hat{\Lambda}_0(t_i)}$. The difference in the log-likelihood functions calculated for the fitted (T) and saturated models is given by:

$$LLd = \sum_{h\in T}\sum_{\mathbf{X}_i\in h}\delta_i \log\left(\frac{\delta_i}{\hat{\Lambda}_0(t_i)}\right) - \delta_i\log\theta_h + \delta_i - \hat{\Lambda}_0(t_i)\theta_h. \tag{12}$$

4. Evolutionary Induction

The evolutionary algorithm described here builds upon the global induction of standard decision trees [19] and extends the method proposed in [18] by introducing various loss functions. In this section, we focuses on the essential elements pertinent to survival analysis, excluding more general aspects. The general schema of evolutionary induction of survival trees in presented in Figure 2.

4.1. Representation and Initialization

In non-terminal nodes, only oblique tests based on hyperplanes are allowed. This means that if categorical features are included in the datasets, they must first be converted, usually into binary features. At each leaf of the survival trees, a Kaplan–Meier (KM) estimator is placed based on the training data that have reached that leaf. As the tree structure evolves during induction, it is crucial to keep track of the locations of all training data points. From a computational standpoint,

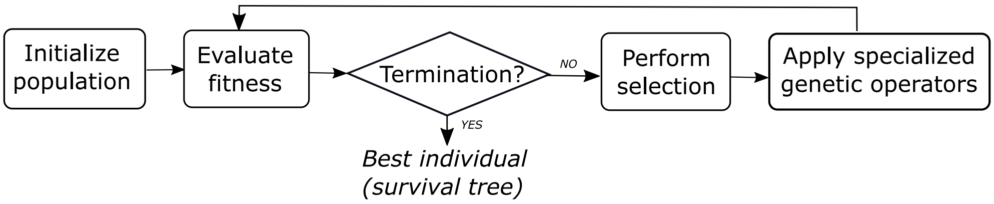

Fig. 2. Schema of evolutionary induction of survival trees.

it is efficient to store this information in a smartly allocated table. Since modifications are often local, only the corresponding areas or subtrees need to be updated. Therefore, individuals are not encoded but are represented as standard binary trees with additional data structures.

Effective population initialization can significantly shorten the evolution process and lead to better resource utilization. The initial trees should be diverse and ideally similar in size to the target survival trees, which are typically compact. A straightforward top-down algorithm with a tree height limit is employed, activated on random small subsets of the training data. Tests in internal nodes are created using randomly selected pairs of observations, known as "dipoles." Dipoles can be formed between uncensored observations or between earlier uncensored observations and later censored ones. When generating tests, longer dipoles are favored, considering the difference in failure times, as their intersection implies that these observations will be in two separate subtrees, eventually becoming leaves. The same mechanism of selecting dipoles and creating tests based on them is also used in the mutation operator.

The evolutionary induction process concludes when, after a specified number of iterations (default 1000 iterations), no individual with a superior fitness value is discovered, or when the maximum number of iterations is reached (default 5000).

4.2. Genetic Operators

In the evolutionary algorithm, two genetic operators are employed, following the standard approach [19]. The first is crossover, which enables the exchange of genetic material between two individuals. In the typical scenario, two nodes (including their subtrees) are randomly chosen from two trees, and their entire subtrees are swapped. Alternatively, it's possible to exchange the tests themselves or perform a crossover with the best-performing individual so far. However, since crossover can sometimes harm the structure of the trees [8], it's used with a relatively low probability (default 0.2).

Mutation, on the other hand, plays a crucial role in diversifying individuals and is applied with a higher probability (default 0.8). This operator can directly modify the tree structure by either pruning a randomly selected subtree down to a leaf or replacing a leaf with an internal node containing a new test. Indirectly, the structure can change when an existing test is altered, such as by randomly adjusting or resetting a weight in the hyperplane. A key aspect for efficient exploration of the search space is the generation of new tests. Here, the dipole mechanism previously described is utilized, allowing the search to proceed sensibly by avoiding ineffective tests.

4.3. Fitness Function

The fitness function is a vital part of every evolutionary algorithm. In evolutionary machine learning, we cannot directly define functions because the goal is to make accurate predictions on unseen data, not just on the training set. To achieve this, we optimize a measure of how well our model performs on the training data, along with a factor that considers the size of the model to avoid overfitting.

For survival tree, T, we calculate the loss function (Loss) and the tree size (number of leaves), and then define the fitness function like this:

$$Fitness(T) = Loss(T) + \alpha(Size(T) - 1), \tag{13}$$

In this study, we explore two types of loss functions: the integrated Brier score (IBS) and the likelihood-based loss (LLd). Adjusting the α value allows us to control the complexity of the resulting tree. Since the two loss functions have different value ranges, the α values considered for each will also differ. In this framework, all tests are treated equally important, irrespective of the number of features they incorporate. If we prioritize simpler tests, we can make the $Size$ term dependent on the number of features used.

5. Experimental Validation and Discussion

In this section, we assessed the results of evolutionary induced oblique survival trees (EIOST) for two different loss functions (see Equation 13): the integrated Brier score and the likelihood-based loss. We evaluated these using two performance measures: IBS and HCI calculated as the mean values over 5 runs of 10-fold cross-validation. The experiments were conducted on five publicly available medical datasets: primary biliary cirrhosis data (pbc418) [7], follicular cell lymphoma study (follic) [25], data from the National Wilm's Tumor Study (nwtco) [27], monoclonal gammopathy data (mgus2) [21], and systolic heart failure data (peakv02) [12]. These datasets had varying percentages of censored cases (ranging from 30.4% to 87.3%), numbers of observations (ranging from 418 to 2231), and attributes (ranging from 4 to 39). A detailed description of the datasets is given in Table 1.

The parameters of the evolutionary algorithm were kept constant during all experiments (population size: 50, maximum number of iterations: 5000, selection pressure: 1.2, minimum number of objects in a leaf: 5). The differences between KM survival functions were assessed using the log-rank test. A significance level of 0.05 was applied for all comparisons.

In Figure 3, we observe the relationship between the predictive ability of EIOST, measured by IBS, and the α parameter ranging from 0.0001 to 0.1. The results are averaged over 5 runs of a 10-fold cross-validation procedure for five medical datasets. In Figure 3a, the IBS values are plotted, showing higher values for α closer to the boundary of its range. The lowest values, indicating better predictive ability, are typically found between 0.001 and 0.1. Figure 3b shows the evaluation of the same tree models using HCI. Unlike the IBS measure, the best-performing model tends to have the highest HCI. The best models, based on HCI, are observed for α values ranging from 0.0002 to 0.001.

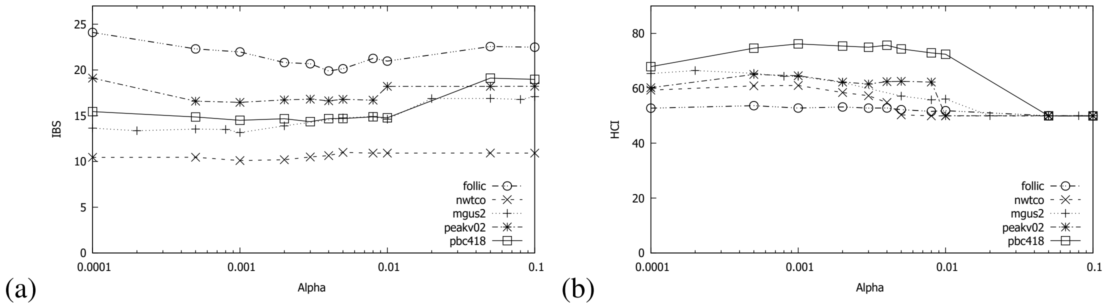

Fig. 3. Dependence of the predictive ability of EIOST with the loss defined as IBS on the α parameter; predictive ability calculated with the use of a) IBS, b) HCI multiplied by 100.

Figure 4 illustrates the relationship between the predictive ability of EIOSTs induced with the LLd loss function and the α parameter. Since the LLd values obtained are considerably larger than the IBS values, the α values controlling model complexity should also be larger than those used for the IBS loss. In our experiments, we consider a range between 0.125 and 32 for α. Similar to the IBS loss, we observe poorer predictive ability for models induced with both low and high values of the complexity parameter, as measured by both IBS and HCI. The optimal range of α for the analyzed datasets is between 3 and 8 for IBS (Figure4a) and between 1 and 3 for HCI (Figure4b).

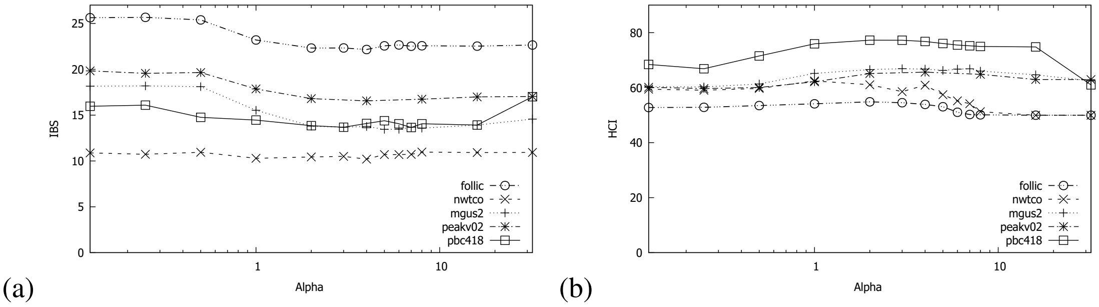

Fig. 4. Dependence of the predictive ability of EIOST with the loss defined as LLd on the α parameter; predictive ability calculated with the use of a) IBS, b) HCI multiplied by 100.

Figure 5 depicts the number of nodes in EIOSTs obtained using the IBS loss (a) and the LLd loss (b) across successive α values. In both figures, we observe a reduction in the number of leaves as the α values increase.

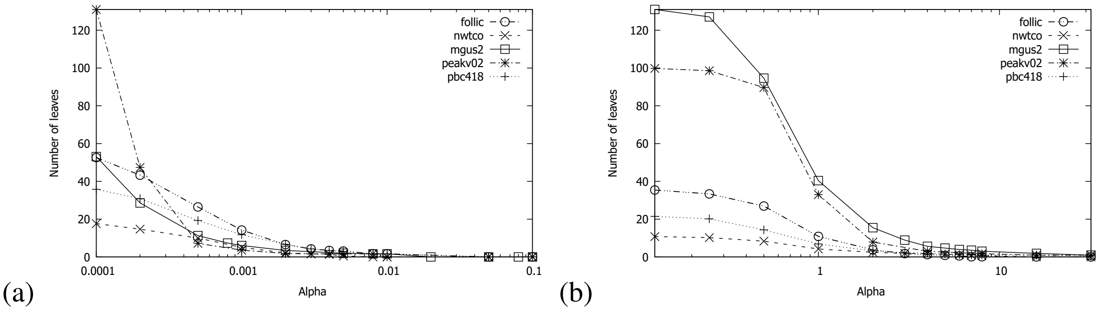

Fig. 5. Dependence of the EIOST size on the α parameter with the loss defined as a) IBS, b) LLd.

In Table 1, we provide a more detailed description of the tree models selected for two loss functions, IBS and LLd, using two evaluation metrics, IBS and HCI. For the nwtco dataset with the IBS loss and the peakv02 dataset with the LLd loss, the models selected using both IBS and HCI metrics are the same. In other cases, the models chosen with IBS tend to be smaller than those selected with HCI, suggesting that higher HCI values are typically associated with more complex trees. Comparing the results for the two loss functions, we notice that HCI values are consistently better for the LLd loss compared to the IBS loss. However, when considering the IBS metric, in 4 out of 5 datasets, the IBS obtained for the LLd loss is worse than for the IBS.

The models reported here were chosen based on the best metric values, separately for IBS and HCI. It is worth noting that the curves in Figures 3 and 4 are often relatively flat over certain ranges of α, indicating that models with similar α values may not differ significantly in the context of predictive ability. Therefore, incorporating knowledge from both metrics might be beneficial in selecting the best tree for a given loss function. To achieve this, we introduced an additional metric called SRank, which is the sum of ranks associated with IBS and HCI metrics separately. The models with the smallest SRank index were selected, providing the best possible values for both metrics simultaneously. The results obtained using the SRank

Table 1. Description of models selected with the use of three metrics: IBS, HCI and SRank for five datasets inducted with the use of IBS and LLd loss functions. M denotes the number of observations, N - the number of attributes, Size denotes the mean number of leaves and IBS (HCI) denotes the mean value IBS (HCI) calculated over 5 runs of 10-fold cross-validation multiplied by 100.

$Loss$		follic	nwtco	mgus2	pekav02	pbc418
	M	541	668	1384	2231	418
	N	4	5	6	39	8
	$\%$ censored	49.7	87.3	30.4	67.5	61.5
IBS	metrics: IBS	19.86	10.1	13.16	16.46	14.35
	α	0.04	0.001	0.001	0.01	0.003
	Size	3.1	4.1	4.7	3.2	3.6
	metrics: HCI	53.66	61.01	66.45	65.03	76.16
	α	0.0005	0.001	0.002	0.0005	0.001
	Size	17.2	4.1	18.5	5.4	8.2
LLd	metrics: IBS	22.16	10.2	13.45	16.56	13.65
	α	4	4	5	4	7
	Size	2.3	2.9	6	4.2	2.1
	metrics: HCI	54.84	62.45	66.89	65.63	77.26
	α	2	1	3	4	3
	Size	5	5.5	10.5	4.2	3.4
IBS	metrics: SRank	4	2	3	3	3
	α	0.004	0.001	0.0002	0.001	0.001
	Size	3.1	4.1	18.5	3.2	8.2
LLd	metrics: SRank	3	3	5	2	4
	α	2	1	3	4	3
	Size	5	5.5	10.5	4.2	3.4

measure are shown in Table 1 under rows labeled as metrics:SRank, where the best sum of ranks is reported. Interestingly, the results for the LLd loss function using the SRank metrics are identical to those for HCI.

In Figures 6, 7, and 8, we observe Kaplan-Meier survival functions derived from survival trees induced with α values chosen based on the SRank measure (Table 1) for the follic, pbc418, and peakv02 datasets. Graphs labeled (a) depict survival trees generated using the IBS loss function, while those labeled (b) depict trees generated using the LLd loss function. It is worth noting, that the survival functions obtained from each solution differ significantly (p-value < 0.05). Despite this, it is visible that the solutions obtained using the LLd loss function appear to be superior for all three datasets.

For the follic dataset, the survival functions associated with leaf numbers 2 (L2) and 3 (L3) in Figure 6a are more similar to each other than the survival functions associated with L2 and L3 in Figure 6b, obtained using the LLd loss.

Analyzing the results for the pbc418 dataset (Figure 7), it is apparent that the use of the IBS loss leads to a more complex survival tree, and the obtained functions do not exhibit generalizing abilities compared to the solution obtained using the LLd function (Figure 7b), where only three leaves with distant median survival times were obtained.

For the peakv02 dataset (Figure 8), similar to the follic dataset, the survival functions for L3 and L4 obtained using the IBS loss are similar, while for the solution obtained using the LLd loss (Figure 8b), the KM survival functions are characterized by more distant median survival

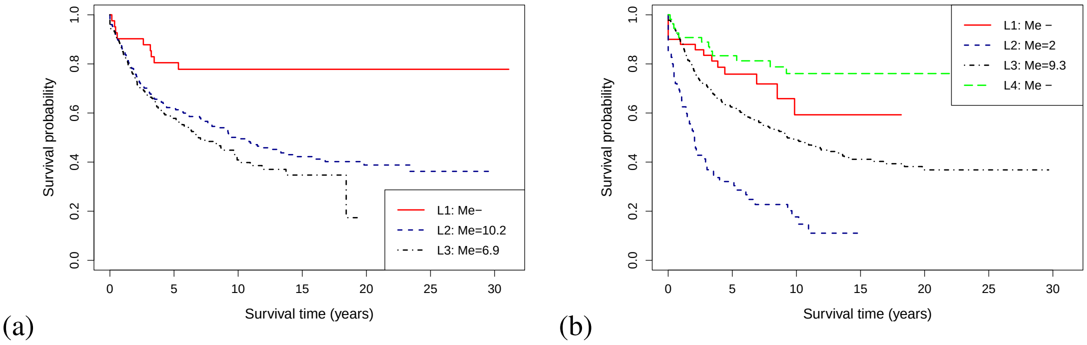

Fig. 6. KM survival functions for trees inducted for the follic dataset with (a) IBS loss function (b) LLd loss function selected by the SRank measure.

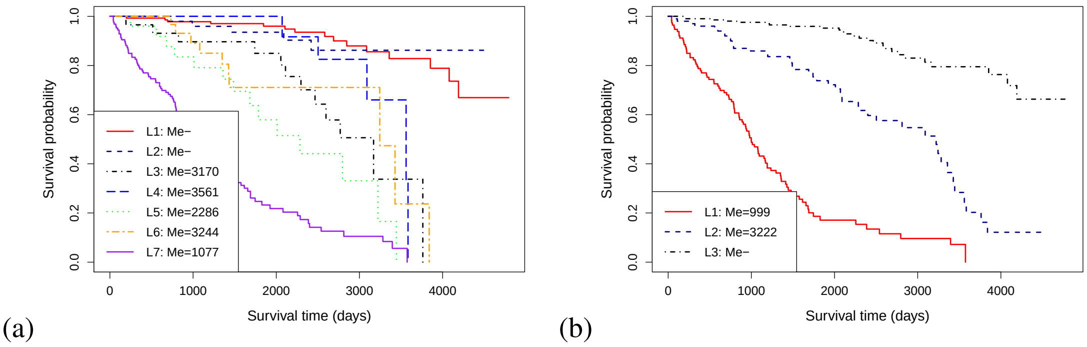

Fig. 7. KM survival functions for trees inducted for the pbc418 dataset with (a) IBS loss function (b) LLd loss function selected by the SRank measure.

times: 4.05 and 8.85 years for L1 and L2, respectively.

Regarding induction times, for the largest dataset analyzed (peakv02), the time required was on the order of several minutes for both IBS (up to 3 minutes) and LLd (up to 8 minutes). We briefly considered incorporating HCI into the fitness function; however, we abandoned this idea due to significantly prolonged induction times. Computing HCI involves considering all pairs of observations each time, resulting in quadratic complexity, which is computationally expensive.

6. Conclusions

In the study, we investigated the impact of two loss functions incorporated into fitness functions of evolutionary algorithm for survival tree induction. We examined the integrated Brier score (IBS), commonly used for evaluating survival models, and the likelihood-based difference between the obtained and saturated model (LLd). Our experiments involved five medical datasets with varying percentages of censored observations.

For each loss function, we generated a set of different models with additional complexity parameter, α. To select the model with the best generalization ability, we used two metrics: the integrated Brier score and Harrell's C-index (HCI). Since the resulting models often differed, we introduced a new metric called SRank, which summed the ranks separately for each metric (IBS and HCI). The SRank values and the tree size did not differ significantly between the two analyzed loss functions. Additionally, the Kaplan–Meier survival functions obtained for each solution—IBS and LLd loss—showed significant differences, as indicated by the log-rank test.

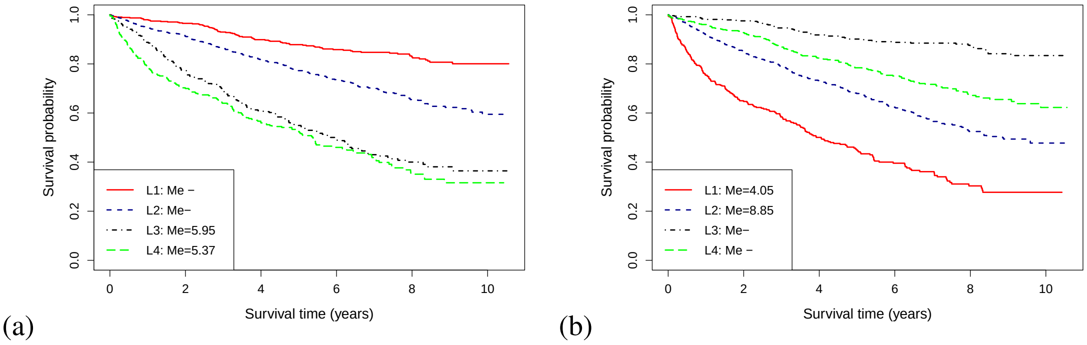

Fig. 8. KM survival functions for trees inducted for the peakv02 dataset with (a) IBS loss function (b) LLd loss function selected by the SRank measure.

Upon analyzing the Kaplan-Meier survival functions associated with the terminal nodes of survival trees induced by the two loss functions, we observed an interesting pattern. In our examples, the survival models generated with the LLd loss function appeared to be more interpretable and therefore more practically useful. Even when accounting for an equal number of leaves, the feature space areas obtained by LLd loss exhibited more varied survival functions, which was not consistently observed with the IBS metric. This may suggests that LLd loss is better at splitting the feature space, resulting in areas that differ from each other in terms of survival experience. Furthermore, it is noteworthy that employing the SRank metric alongside the LLd loss function yielded outcomes equivalent to those chosen by HCI alone.

In this paper, we analyzed survival data with right censoring and continuous survival time. We aim to extend our method to accommodate other types of survival data, such as discrete survival data and data with competing risks.

Acknowledgements

This work was supported by Bialystok University of Technology under the grant WZ/WI-IIT/4/2023 founded by Ministry of Science and Higher Education.

References

1. Aalen, O.: Nonparametric Inference for a Family of Counting Processes. Ann. Stat. 6 (4), 701-726 (1978)
2. Bertsimas, D., Dunn, J., Gibson, E., Orfanoudaki, A.: Optimal Survival Trees. Mach. Learn. 111 (8), 2951-3023 (2022)
3. Breiman, L., Friedman, J., Olshen, R., Stone, C.: Classification and Regression Trees. Wadsworth, Belmont, CA (1984)
4. Ciampi, A., Thiffault, J., Nakache, J.P., Asselain, B.: Stratification by Stepwise Regression, Correspondence Analysis and Recursive Partition: A Comparison of Three Methods of Analysis for Survival Data with Covariates. Comput Stat Data Anal 4 (3), 185-204 (1986)
5. Cox, D.R.: Regression Models and Life Tables (with discussion). J. R. Stat. Soc. Ser. B 34, 187-220 (1972)
6. Davis, R.B., Anderson, J.R.: Exponential Survival Trees. Stat. Med. 8, 947-961 (1989)
7. Fleming, T.R., Harrington, D.P.: Counting Processes and Survival Analysis. John Wiley & Sons (1991)
8. Freitas, A.A.: Data Mining and Knowledge Discovery with Evolutionary Algorithms.

Springer Science & Business Media (2002)

9. Graf, E., Schmoor, C., Sauerbrei, W., Schumacher, M.: Assessment and Comparison of Prognostic Classification Schemes for Survival Data. Stat. Med. 18, 2529-2545 (1999)
10. Harrell, F.E., Califf, R.M., Pryor, D.B., Lee, K.L., Rosati, R.A.: Evaluating the Yield of Medical Tests. JAMA 247, 2543-2546 (1982)
11. Hothorn, T., Hornik, K., Zeileis, A.: Unbiased recursive partitioning: A Conditional Inference Framework. Comput. Graph. Stat. 15 (3), 651-674 (2006)
12. Hsich, E., Gorodeski, E.Z., Blackstone, E.H., Ishwaran, H., Lauer, M.S.: Identifying Important Risk Factors for Survival in Patient with Systolic Heart Failure Using Random Survival Forests. Circulation: Cardiovascular Quality and Outcomes 4 (1), 39-45 (2011)
13. Jaeger, B.C., Long, D.L., Long, D.M., Sims, M., Szychowski, J.M., Min, Y.I., Mcclure, L.A., Howard, G., Simon, N.: Oblique Random Survival Forests. Ann. Appl. Stat. 13 (3), 1847-1883 (2019)
14. Jaeger, B. C., Sawyer, W., Kristin, L., Speiser, J.L., Segar, M.W., Ambarish P., Pajewski, N.M.: Accelerated and Interpretable Oblique Random Survival Forests. J. Comput. Graph. Stat. 33 (1), 192-207 (2024)
15. Kalbfleisch, J.D., Prentice, R.L.: The Statistical Analysis of Failure Time Data. John Wiley & Sons (2011)
16. Kaplan, E.L., Meier, P.: Nonparametric Estimation from Incomplete Observations. JASA 53, 457-481 (1958)
17. Kretowska, M.: Piecewise-Linear Criterion Functions in Oblique Survival Trees Induction. Artif. Intell. Med. 75, 32-39 (2017)
18. Kretowska, M., Kretowski, M.: Global Induction of Oblique Survival Trees. In: Franco, L. et al. (eds.): Computational Science - ICCS 2024, LNCS 14835, 379-386 (2024)
19. Kretowski, M.: Evolutionary Decision Trees in Large-scale Data Mining. Springer (2019)
20. Kundu, M.G., Ghosh, S.: Survival Trees Based on Heterogeneity in Time-to-Event and Censoring Distributions Using Parameter Instability Test. Statistical Analysis and Data Mining: The ASA Data Science Journal 14 (5), 466-483 (2021)
21. Kyle, R.A., Therneau, T.M., Rajkumar, S.V., Offord, J.R., Larson, D.R., Plevak, M.F., Melton III, L.J.: A Long-term Study of Prognosis in Monoclonal Gammopathy of Undetermined Significance. N. Engl. J. Med. 346 (8), 564-569 (2002)
22. LeBlanc, M., Crowley, J.: Relative Risk Trees for Censored Survival Data. Biometrics 48, 411-425 (1992)
23. LeBlanc, M., Crowley, J.: Survival Trees by Goodness of Split. JASA 88 (422), 457-467 (1993)
24. Nelson, W.: Theory and Applications of Hazard Plotting for Censored Failure Data. Technometrics 14 (4), 945-966 (1972)
25. Pintilie, M.: Competing Risks: A Practical Perspective, John Wiley & Sons (2006)
26. Segal, M.R.: Regression Trees for Censored Data. Biometrics 44, 35-47 (1988)
27. Therneau, T.M.: Survival: Survival Analysis, R package version 2.39 (2016)
28. Therneau, T.M., Grambsch, P.M., Fleming, T.R.: Martingale-based Residuals for Survival Models. Biometrika 77 (1), 147-160 (1990)
29. Wang, H., Chen, X., Li, G.: Survival Forests with R-Squared Splitting Rules. J. Comput. Biol. 25 (4), 388-395 (2018)
30. Wang, H., Zhou, L.: Random Survival Forest with Space Extensions for Censored Data. Artif. Intell. Med. 79, 52-61 (2017)
31. Wang, P., Li, Y., Reddy, C.K.: Machine Learning for Survival Analysis: A Survey. ACM Computing Survey 51 (6), 1-36 (2019)