Artificial Intelligence in Medicine 75 (2017) 32–39

Contents lists available at ScienceDirect

Artificial Intelligence in Medicine

journal homepage: www.elsevier.com/locate/aiim

Piecewise-linear criterion functions in oblique survival tree induction

Malgorzata Kretowska

Faculty of Computer Science, Bialystok University of Technology, Wiejska 45a, 15-351 Bialystok, Poland

ARTICLE INFO

Article history:
Received 15 July 2016
Received in revised form 7 November 2016
Accepted 28 December 2016

Keywords:
Piecewise-linear criterion function
Survival tree
Oblique splits
Right-censored data

ABSTRACT

Objective: Recursive partitioning is a common, assumption-free method of survival data analysis. It focuses mainly on univariate trees, which use splits based on a single variable in each internal node. In this paper, I provide an extension of an oblique survival tree induction technique, in which axis-parallel splits are replaced by hyperplanes, dividing the feature space into areas with a homogeneous survival experience.
Method and materials: The proposed tree induction algorithm consists of two steps. The first covers the induction of a large tree with internal nodes represented by hyperplanes, whose positions are calculated by the minimization of a piecewise-linear criterion function, the dipolar criterion. The other phase uses a split-complexity algorithm to prune unnecessary tree branches and a 10-fold cross-validation technique to choose the best tree. The terminal nodes of the final tree are characterised by Kaplan–Meier survival functions. A synthetic data set was used to test the performance, while seven real data sets were exploited to validate the proposed method.
Results: The evaluation of the method was focused on two features: predictive ability and tree size. These were compared with two univariate tree models: the conditional inference tree and recursive partitioning for survival trees, respectively. The comparison of the predictive ability, expressed as an integrated Brier score, showed no statistically significant differences ($p = 0.486$) among the three methods. Similar results were obtained for the tree size ($p = 0.11$), which was calculated as a median value over 20 runs of a 10-fold cross-validation.
Conclusions: The predictive ability of trees generated using piecewise-linear criterion functions is comparable to that of univariate tree-based models. Although a similar conclusion may be drawn from the analysis of the tree size, in the majority of the studied cases, the number of nodes of the dipolar tree is one of the smallest among all the methods.

1. Introduction

The prediction of failure time is one of the major tasks in survival analysis. In the medical domain, it often describes the time to death or disease relapse. Cox's proportional hazards model [1] is one of the most common statistical methods used to analyse survival data. This semi-parametric model requires the fulfillment of certain assumptions about an analysed phenomenon that is often difficult to achieve. Some other restrictions concern accelerated failure time models [2], for which the analytical form of the relationship between the survival function and the covariates should be established. The requirements accompanying statistical models result in the development of alternative, assumption-free methods of survival analysis. Among them, tree-based models play an important role.

Survival trees are mainly intended to analyze right-censored data, and their first applications appeared in the eighties. As pointed out by LeBlanc and Crowley [3], tree induction algorithms may be categorized from the point of view of a splitting criterion (i.e., the impurity or the between-node separation measure). The first group covers the algorithms following the CART (Classification and Regression Trees) methodology [4]. Gordon and Olshen [5] used the Wasserstein metric, Davis and Anderson [6] applied exponential log-likelihood loss, LeBlanc and Crowley [7] applied an approximation of the full likelihood for the proportional hazards model, while Therneau et al. [8] used martingale-based residuals from the Cox model. The other group of algorithms is usually based on the Tarone–Ware class of two-sample statistics for censored data, such as the log-rank test [9–11].

Another important aspect of tree induction methods is a stopping criterion. Its appropriate choice causes the final tree to have a good generalization ability; too small or too large trees lead to an under- or overfitting phenomenon. A common way to select the

E-mail address: m.kretowska@pb.edu.pl

http://dx.doi.org/10.1016/j.artmed.2016.12.004
0933-3657/

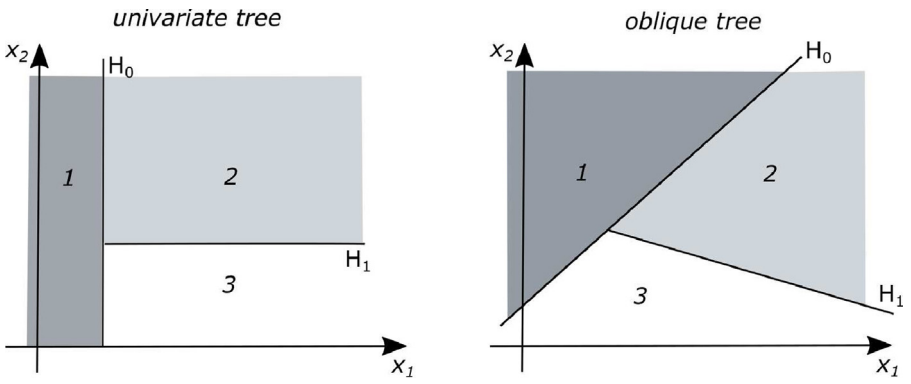


Fig. 1. Division of the feature space into three disjoined areas (1,2,3) by the hyperplanes H_0 and H_1 in univariate and oblique trees.

final tree is to build a large tree and then prune some of its branches. The idea was proposed in [4] as cost-complexity pruning and was then extended to survival trees by LeBlanc and Crowley [11] and referred to as a split-complexity algorithm. Another approach does not separate the pruning phase from the induction process. Rather, a decision on the split importance is made during creation of the node. Hothorn et al. [12] proposed the use of multiple test procedures and to stop the split if the test results are not statistically significant at a given value of α.

Comparisons of different splitting criterion and pruning techniques was presented in [13,14], while a comprehensive overview of tree-based models was provided by Bou–Hamad et al. [15].

Although single trees are now often replaced by more powerful ensembles of trees, they have one undeniable advantage: an insight into data [15], made possible by analysing splits in subsequent internal nodes, which divide the feature space into homogeneous areas. The survival trees are narrowed to univariate trees, in which one split is based on only one variable. In real data, the borders between regions with different survival experiences need not be parallel to the coordinate axes (Fig. 1). If we use a univariate tree to solve this problem, we must create a number of internal nodes instead of one hyperplane.

In this paper, I develop a method of oblique survival tree induction, introduced briefly in [16]. Here, a single split is equivalent to any hyperplane, whose location is determined by the minimization of a convex and piecewise-linear (CPL) criterion function [17] built based on right-censored data. The performance of the final tree, chosen by a split-complexity pruning method [11], was compared with those of two univariate tree-based models: a conditional inference tree [12] and recursive partitioning for survival trees [18] (R package: rpart), which corresponds to a method proposed by LeBlanc and Crowley [7].

The paper consists of 7 sections. Section 2 introduces a definition and basic concepts of survival data. In Section 3, the piecewise-linear criterion function, the dipolar criterion, is presented. An oblique survival tree induction algorithm is described in Section 4. Possible validation measures are presented in Section 5, while Section 6 shows the results of the experiments on synthetic and real data. Section 7 contains the conclusions.

2. Survival data

Observe random variable $P=(\mathbf{X}, T, \Delta)$, where $\mathbf{X}$ is the N-dimensional feature vector, $T=\min(T_0, C)$, T_0 is the survival time with the distribution function f_t, C is the censoring time with the distribution function f_c, and Δ is the censoring indicator $\Delta = I(T_0 < C)$. A learning sample, L, consists of M observations $(\mathbf{x}_i, t_i, \delta_i)$, $i=1, 2, \ldots, M$, where $\mathbf{x}_i$ is the N-dimensional feature vector describing the ith patient, t_i is the survival time, and δ_i is the failure indicator, which takes one of two values: 0 for censored observations or 1 for uncensored ones.

The distribution of the survival time may be described by several functions. One of them is a survival function, which represents the probability of surviving beyond the time t: $S(t)=P(T>t)$. One of the most common nonparametric estimators of the survival function is the Kaplan–Meier product-limit estimator [19]. If we assume that the events of interest occur at D distinct times $t_{(1)} < t_{(2)} < \ldots < t_{(D)}$, the estimator is calculated as follows:

$$\hat{S}(t) = \prod_{j|t_{(j)} \le t} \left(\frac{m_j - d_j}{m_j} \right) \tag{1}$$

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 $t_{(j)}$).

3. Dipolar criterion function

CPL criterion functions are common methods used in data analysis. In this paper, a CPL function, the dipolar criterion $\Psi_d(\cdot)$ [17], was used to determine the splits in the internal nodes of survival trees.

Let us introduce the augmented feature and weight vectors:

$$\begin{aligned} \boldsymbol{z} &= [1, x_1, x_2, \ldots, x_N]^T \\ \boldsymbol{v} &= [-\theta, w_1, w_2, \ldots, w_N]^T \end{aligned} \tag{2}$$

For any feature vector $\boldsymbol{z}_j$, $j=1, 2, \ldots, M$ from the learning set L, we can define two piecewise-linear penalty functions:

$$\varphi_j^+(\boldsymbol{v}) = \begin{cases} \delta_j - \boldsymbol{v}^T\boldsymbol{z}_j & \text{if} \quad \boldsymbol{v}^T\boldsymbol{z}_j \le \delta_j \\ 0 & \text{if} \quad \boldsymbol{v}^T\boldsymbol{z}_j > \delta_j \end{cases} \tag{3}$$

and

$$\varphi_j^-(\boldsymbol{v}) = \begin{cases} \delta_j + \boldsymbol{v}^T\boldsymbol{z}_j & \text{if} \quad \boldsymbol{v}^T\boldsymbol{z}_j \ge -\delta_j \\ 0 & \text{if} \quad \boldsymbol{v}^T\boldsymbol{z}_j < -\delta_j \end{cases} \tag{4}$$

where $\delta_j \ge 0$ is a margin usually equal to 1. In Fig. 2, we can see graphical representations of $\varphi_j^+(\boldsymbol{v})$ and $\varphi_j^-(\boldsymbol{v})$ compared to the scalar product $\boldsymbol{v}^T\boldsymbol{z}_j$.

If we take into account a hyperplane $H(\boldsymbol{v}) = \{\boldsymbol{z} : \boldsymbol{v}^T\boldsymbol{z} = 0\}$ (or, equivalently, $H(\boldsymbol{w}, \theta) = \{\boldsymbol{x} : \boldsymbol{w}^T\boldsymbol{x} = \theta\}$), the functions $\varphi_j^+(\boldsymbol{v})$ and $\varphi_j^-(\boldsymbol{v})$, associated with a given feature vector $\boldsymbol{z}_j$, penalize for the inappropriate position of $H(\boldsymbol{v})$ toward $\boldsymbol{z}_j$. The minimization of the penalty enforces a correct localisation of $H(\boldsymbol{v})$; in addition, with a margin greater than zero, the hyperplane is unable to pass through $\boldsymbol{z}_j$, which improves the generalization ability.

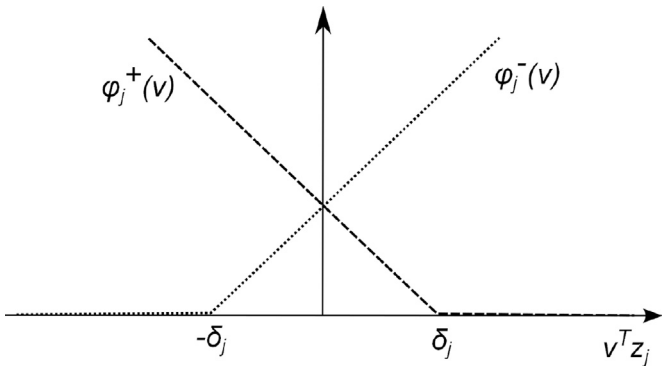


Fig. 2. Penalty functions $\varphi_j^-(\boldsymbol{v})$ and $\varphi_j^+(\boldsymbol{v})$.

In Fig. 3, we can see the areas of the feature space with greater than zero values of penalty functions $\varphi_j^+(\boldsymbol{v})$ (Fig. 3a) and $\varphi_j^-(\boldsymbol{v})$ (Fig. 3b). Minimizing the penalty shifts the hyperplane $H(\boldsymbol{v})$ causing $\boldsymbol{z}_j$ to be placed in areas with penalty functions equal to zero (uncolored).

The dipolar criterion function is based on the features of a dipole, which is a pair of feature vectors $(\boldsymbol{z}_j, \boldsymbol{z}_{j'})$, where $j \neq j', j = 1, 2, \ldots, M$ and $j' = 1, 2, \ldots, M$. We can distinguish two types of dipoles. Mixed dipoles are created between two feature vectors that should be separated; hence, the function associated with the mixed dipole is the sum of two different types of penalty functions:

$$\begin{aligned} \varphi_{jj'}^{m^+}(\boldsymbol{v}) &= \varphi_j^+(\boldsymbol{v}) + \varphi_{j'}^-(\boldsymbol{v}) \quad \text{or} \\ \varphi_{jj'}^{m^-}(\boldsymbol{v}) &= \varphi_j^-(\boldsymbol{v}) + \varphi_{j'}^+(\boldsymbol{v}) \end{aligned} \tag{5}$$

However, pure dipoles are created between two feature vectors that should not be separated. The function associated with a pure dipole is the sum of two penalty functions of the same type:

$$\begin{aligned} \varphi_{jj'}^{p^+}(\boldsymbol{v}) &= \varphi_j^+(\boldsymbol{v}) + \varphi_{j'}^+(\boldsymbol{v}) \quad \text{or} \\ \varphi_{jj'}^{p^-}(\boldsymbol{v}) &= \varphi_j^-(\boldsymbol{v}) + \varphi_{j'}^-(\boldsymbol{v}) \end{aligned} \tag{6}$$

A hyperplane $H(\boldsymbol{v})$ divides a dipole $(\boldsymbol{z}_j, \boldsymbol{z}_{j'})$ if one of the feature vectors ($\boldsymbol{z}_j$ or $\boldsymbol{z}_{j'}$) is situated on the positive side of the hyperplane ($\boldsymbol{v}^T\boldsymbol{z}_j \geq 0$ or $\boldsymbol{v}^T\boldsymbol{z}_{j'} \geq 0$, respectively), while the other is on the negative side.

Depending on the associated function, dipoles may have a positive or negative orientation. A mixed dipole $(\boldsymbol{z}_j, \boldsymbol{z}_{j'})$ has a:

- positive orientation (function $\varphi_{jj'}^{m^+}(\boldsymbol{v})$) if we expect that vector $\boldsymbol{z}_j$ will be situated on the positive side of a hyperplane $H(\boldsymbol{v})$ and $\boldsymbol{z}_{j'}$ on the negative side
- negative orientation (function $\varphi_{jj'}^{m^-}(\boldsymbol{v})$) if we expect that vector $\boldsymbol{z}_{j'}$ will be situated on the positive side of a hyperplane $H(\boldsymbol{v})$ and $\boldsymbol{z}_j$ on the negative side

A pure dipole $(\boldsymbol{z}_j, \boldsymbol{z}_{j'})$ has a:

- positive orientation (function $\varphi_{jj'}^{p^+}(\boldsymbol{v})$) if we expect that two vectors ($\boldsymbol{z}_j$ and $\boldsymbol{z}_{j'}$) will be situated on the positive side of a hyperplane $H(\boldsymbol{v})$
- negative orientation (function $\varphi_{jj'}^{p^-}(\boldsymbol{v})$) if we expect that two vectors ($\boldsymbol{z}_j$ and $\boldsymbol{z}_{j'}$) will be situated on the negative side of a hyperplane $H(\boldsymbol{v})$

With L, we can formulate and apply some rules of dipole creation in order to solve a given issue. For a given set of mixed, pure, and fixed dipole orientations, the dipolar criterion function is defined as follows:

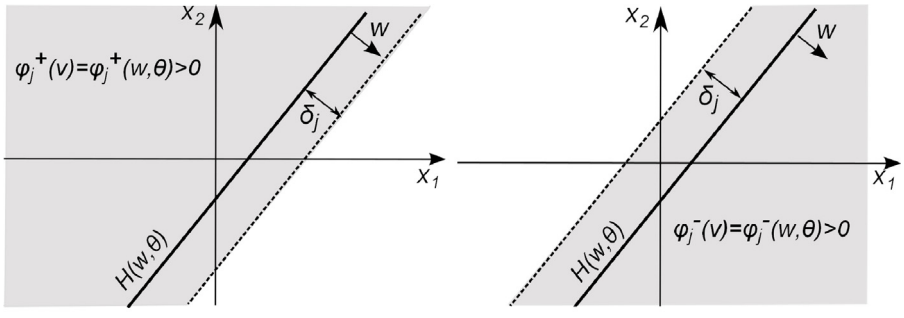
$$\begin{aligned} \Psi_d(\boldsymbol{v}) = &\sum_{(j,j')\in I^{p^+}} \alpha_{jj'}\varphi_{jj'}^{p^+}(\boldsymbol{v}) + \sum_{(j,j')\in I^{p^-}} \alpha_{jj'}\varphi_{jj'}^{p^-}(\boldsymbol{v}) + \sum_{(j,j')\in I^{m^+}} \alpha_{jj'}\varphi_{jj'}^{m^+}(\boldsymbol{v}) \\ &+ \sum_{(j,j')\in I^{m^-}} \alpha_{jj'}\varphi_{jj'}^{m^-}(\boldsymbol{v}) \end{aligned} \tag{7}$$

where I^{p^+} (I^{p^-}) is a set of pure dipoles with a positive (negative) orientation, I^{m^+} (I^{m^-}) is a set of mixed dipoles with a positive (negative) orientation, and $\alpha_{jj'}$ is the price of a dipole $(\boldsymbol{z}_j, \boldsymbol{z}_{j'})$. In the experiments, the value of $\alpha_{jj'}$ was constant and equal to 1 for pure dipoles and 1000 for mixed dipoles. Such a big difference forces the algorithm to divide mixed dipoles, even when the values of associated penalty functions are small in comparison to pure dipoles.

The dipolar criterion function $\Psi_d(\boldsymbol{v})$ is a CPL function; therefore, its minimization may be conducted based on a basis exchange algorithm [20]. The minimization process consists of the following steps:

1. For the given dipole orientation, find the minimum of $\Psi_d(\boldsymbol{v})$: $\Psi_d(\boldsymbol{v}^*)$.
2. Change the dipole orientation to reduce $\Psi_d(\boldsymbol{v}^*)$: $\Psi_d^{new}(\boldsymbol{v}^*)$.
3. If $\Psi_d^{new}(\boldsymbol{v}^*) < \Psi_d(\boldsymbol{v}^*)$, return to step 3; otherwise, the minimization process is over.

As a result, we receive the hyperplane $H(\boldsymbol{v}^*)$, which, informally speaking, divides a possibly high amount of mixed dipoles and a possibly low amount of pure dipoles.



(a) (b)

Fig. 3. The areas of the feature space (gray) with greater than zero values of the penalty functions $\varphi_j^-(\boldsymbol{v})$ and $\varphi_j^+(\boldsymbol{v})$.

4. Oblique survival tree

A tree is a structure consisting of internal and terminal nodes. Trees divide a feature space (internal nodes) in order to obtain homogeneous areas of data (terminal nodes) in a given task. In a binary tree, each internal node has two child nodes; terminal nodes, also called leaves, have no child nodes. They represent the obtained data areas.

The dipolar survival tree is a binary, oblique tree, where tests in internal nodes take the form of a hyperplane, $H(\boldsymbol{v})$. The hyperplane, which is associated with a root node, divides the feature space into two sub-spaces. The first sub-space consists of feature vectors placed on the positive side of the hyperplane ($\boldsymbol{v}^T\boldsymbol{z}_j \geq 0$), while the other features vectors situated on its negative side ($\boldsymbol{v}^T\boldsymbol{z}_j < 0$). The obtained sub-spaces are then divided in similar manner by the hyperplane associated with successive internal nodes: positive sub-space by the left child node and the negative one by the right child node. The terminal nodes contain the feature vectors that belong to the obtained feature space areas. In survival data, each terminal node is characterised by the Kaplan–Meier survival function, as estimated based on the feature vectors that have reached the node (Fig. 4).

4.1. Dipolar survival tree induction

Considering continuous survival times, our aim was to separate the feature vectors for which the difference between the survival times is large, while vectors with similar survival times were not separated. To judge whether a pair of vectors from a learning set, L, should form a mixed or pure dipole, we introduced a set of survival time differences, D. Assuming that $D=\emptyset$, $i = 1, \ldots, M$, and $j = i+1, \ldots, M$, D is as follows:

- If $\delta_i = \delta_j = 1$, then $D = D \cup \{|t_i - t_j|\}$.
- If $\delta_i = 0$, $\delta_j = 1$, and $t_i > t_j$, then $D = D \cup \{t_i - t_j\}$.
- If $\delta_i = 1$, $\delta_j = 0$, and $t_i < t_j$, then $D = D \cup \{t_j - t_i\}$.

The elements of $D = \{d_k\}, k = 1, 2, \ldots, K$ were then sorted in ascending order, creating a sequence of values:

$$d_{(1)} \leq d_{(2)} \leq \ldots \leq d_{(K)} \tag{8}$$

This led to the following rules of dipole formation:

1. A pair of feature vectors ($\mathbf{x}_i$, $\mathbf{x}_j$) forms a pure dipole if
   - $\delta_i = \delta_j = 1$ and $|t_i - t_j| < d_{(\lfloor \eta^* K \rfloor)}$.
2. A pair of feature vectors ($\mathbf{x}_i$, $\mathbf{x}_j$) forms a mixed dipole if
   - $\delta_i = \delta_j = 1$ and $|t_i - t_j| \geq d_{(\lfloor \zeta^* K \rfloor)}$
   - $\delta_i = 0$, $\delta_j = 1$, $t_i > t_j$, and $t_i - t_j \geq d_{(\lfloor \zeta^* K \rfloor)}$
   - $\delta_i = 1$, $\delta_j = 0$, $t_i < t_j$, and $t_j - t_i \geq d_{(\lfloor \zeta^* K \rfloor)}$

where η and ζ are parameters and $\lfloor x \rfloor$ is the floor function, which returns the largest integer not greater than x.

Hence, pure dipoles are created only between two uncensored observations since, in other cases, we cannot be sure that the difference between two survival times will be small enough to fulfill the condition $|t_i - t_j| < d_{(\lfloor \eta^* K \rfloor)}$. In mixed dipoles, one of the observations may be censored, but only if its survival time is greater than the survival time of the uncensored case. If the time difference is greater than or equal to $d_{(\lfloor \zeta^* K \rfloor)}$, a mixed dipole is created.

The values of $d_{(\lfloor \eta^* K \rfloor)}$ and $d_{(\lfloor \zeta^* K \rfloor)}$ are established at the beginning of the tree induction algorithm and hold true during the whole process. Taking into account the ith internal node and assuming that a learning set L_i contains feature vectors that have reached the ith node, the following steps for node induction can be derived:

1. Based on the learning set L_i, create sets of mixed and pure dipoles.
2. If the set of mixed dipoles is empty, then the ith node becomes a terminal one, and we finish the algorithm.
3. Initialize a value of $\boldsymbol{v}_i = \boldsymbol{v}_{i_0}$.
4. Minimize the dipolar criterion function $\Psi_d(\boldsymbol{v}_{i_0})$ in order to receive $\boldsymbol{v}_{i_{\min}}$.
5. For each $\boldsymbol{z}_j \in L_j$, $j = 1, 2, \ldots, M_i$, calculate $h_j = \boldsymbol{v}_{i_{\min}}^T \boldsymbol{z}_j$.
6. Create two sets of learning samples: $L_{i_{left}} = \{\boldsymbol{z}_j : h_j \geq 0\}$ and $L_{i_{right}} = \{\boldsymbol{z}_j : h_j < 0\}, j = 1, 2, \ldots, M_i$, with a cardinality of $M_{i_{left}}$ and $M_{i_{right}}$ respectively.
7. If $M_{i_{left}} < 10$ or $M_{i_{right}} < 10$, then the ith node becomes a terminal one, and we finish the algorithm.
8. Repeat steps 1–8 for data sets $L_{i_{left}}$ and $L_{i_{right}}$ to create the left and right node, respectively.

4.2. Pruning and selection of the final tree

Tree induction algorithms usually build trees that overfit the learning data, which leads to poor generalization. To improve it, we used the pruning algorithm proposed by LeBlanc and Crowley [11]. The method, derived from the CART pruning algorithm [4] for a given tree, T, maximizes a split-complexity measure, defined as follows:

$$G_\alpha(T) = G(T) - \alpha|T| \tag{9}$$

where α is a non-negative complexity parameter, $|T|$ is the number of internal nodes of T, and $G(T)$ is the sum of the standardized splitting statistics $G(t)$ calculated over all the internal nodes of T:

$$G(T) = \sum_{t \in In(T)} G(t) \tag{10}$$

where $In(T)$ is the set of internal nodes of T and $G(t)$ represents the log-rank statistics with χ^2_1 distribution. We found that the choice of the best subtree (i.e., the tree that maximizes (9)) is strictly related to the value of α. Although α is a continuous variable, the number of subtrees of T is finite. Therefore, only one subtree is the best for the whole interval of α values.

The first phase of the algorithm creates a sequence of subtrees associated with increasing values of α. A pair (T_i, α_i) means that T_i is the best subtree for $\alpha \in [\alpha_i, \alpha_{i+1})$. For $\alpha_0 = 0$, the choice of subtree was obvious ($T_0 = T$), but to find the next values of $\alpha > 0$, for each internal node $t \in In(T_0)$, we calculated

$$\gamma_t = \frac{T_t}{|T_t|} \tag{11}$$

where T_t is a branch of T_0 with a root node in t. The next value of α is $\alpha_1 = \min_{t \in In(T_0)}\{\gamma_t\} = \gamma_{t^*}$. The tree associated with α_1 is $T_1 = T_0 - T_{t^*}$ (i.e., T_0 pruned in the node t^*). The procedure was then repeated for subtrees T_1, T_2, ... until we received only the root node. As a result, the nested sequence of trees with corresponding α values obtained as follows:

$$\begin{aligned} &T_0 \succ T_1 \succ \ldots \succ T_m \\ &0 < \alpha_1 < \ldots < \infty \end{aligned} \tag{12}$$

where T_0 is an unpruned tree, T_m is the root node, and $T_i \succ T_{i+1}$ means that T_{i+1} is a subtree of T_i. The choice of the subtree with the best generalization was narrowed to the following set: $T_0, T_1, \ldots, T_m$.

To decide which of the trees from the sequence should be the final tree, a K-fold cross-validation was applied. In the method, the learning set L was divided into K folds ($F_1, F_2, \ldots, F_K$) with the same observation number. In the kth iteration, $k = 1, 2, \ldots, K$, fold F_k is a test sample, while the other folds constitute the kth learning sample $L_k = L - F_k$. In each iteration, we induced the tree based on the

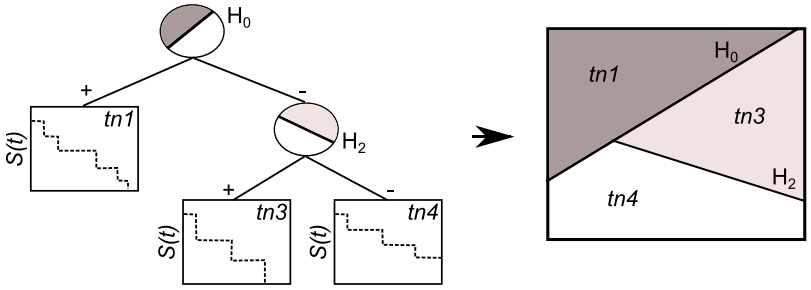


Fig. 4. A survival tree. Internal nodes divide the feature space into disjoined areas and terminal nodes, characterised by Kaplan–Meier survival functions, represent the distinguished areas.

Table 1
Description of synthetic data set; # obs denotes the number of observations, # events denotes the number of failures.

x	y	# obs	# events	Median survival time
$N(2.5,1.2)$	$N(4,1.2)$	100	93	18.8
$N(4.5,1.2)$	$N(6,1.2)$	100	94	7.1
$N(5,1.2)$	$N(2,1.2)$	100	83	75.7
$N(7,1.2)$	$N(4,1.2)$	100	91	60.5

learning set L_k and pruned it with the complexity parameter (9) equal to $\beta_l = \sqrt{\alpha_l \alpha_{l+1}}$, $l = 1, \ldots, m-1$. For each obtained subtree, we calculated $G_{(k,l)} = G_{\alpha_c}(T)$, where α_c is a penalty equal to 2.07, which corresponds to the 0.15 significance level of the χ^2_1 distribution [11]. $G_{(k,l)}$, calculated with the use of test sample F_k, may be interpreted as the quality of the subtree obtained in the kth iteration with the complexity parameter β_l.

Taking the average of $G_{(k,l)}$ over all the folds for each β_l as follows

$$\bar{G}_{(\cdot,l)} = \frac{1}{K} \sum_{k=1,\ldots,K} G_{(k,l)} \tag{13}$$

we chose the subtree that maximizes $\bar{G}_{(\cdot,l)}$.

5. Model validation

The assessment of the predictive ability of the model in terms of survival data is less straightforward than that of regression models. Classical measures (i.e., the mean squared error or the mean absolute error (MAE)) require knowledge of the target values. In survival data, such values are available only for uncensored observations; for censored cases, the exact failure times are unknown. Hence, the proposed validation measures of the survival models are based on the difference between the survival functions [21] or on the ordering of the predicted and real survival times [22].

One of the most common measures used for the assessment of survival models is an integrated Brier score (IBS) [21]. For a fixed time point, t, the observations are divided into three groups:

1. $t_i \le t$ 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. $t_i > t$ 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. $t_i \le t$ 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 derived from group 3 do not contribute to the Brier score, the loss of information should be compensated by the additional weighting of the existing contributions. The observations derived from group 1 have the weight $\hat{G}(t_i)^{-1}$, and those derived from group 2 the weight $\hat{G}(t)^{-1}$, where $\hat{G}(t)$ denotes the Kaplan–Meier estimator of the censoring distribution. It is calculated based on observations $(t_i, 1-\delta_i)$. The definition of the Brier score is given as follows:

$$BS(t) = \frac{1}{n} \sum_{i=1}^{N} (\hat{S}(t|\mathbf{x}_i)^2 I(t_i \le t \wedge \delta_i = 1) \hat{G}(t_i)^{-1} + (1 - \hat{S}(t|\mathbf{x}_i))^2 I(t_i > t) \hat{G}(t)^{-1}) \tag{14}$$

where $I(condition)$ is equal to 1 if the condition is fulfilled and zero otherwise.

The IBS is calculated as follows:

$$IBS = \frac{1}{\max(t_i)} \int_0^{\max(t_i)} BS(t) dt \tag{15}$$

Harrel's c-index [22] is one of the most common ranked measures of predictive ability. For its calculation, we need to define all the evaluable pairs of observations. In survival data, the evaluable pair is a pair of observations in which we can establish the order of failure occurrence. From this set, we excluded pairs of censored observations and pairs of censored and uncensored observations in which the censored time was less than the failure time. The c-index is a ratio of concordant pairs (in which the order of the predicted and real failure times is the same) to all evaluable pairs. Other examples of correlation measures include Kendall's τ_a and Somer's D_{yx} [23].

6. Experiments and results

Two experiments were conducted. The first shows the results of the synthetic data set and aimed at a detailed analysis of the proposed method. The second presents the results of seven real data sets. The predictive ability of the oblique tree (DSTree) was compared with two other tree-based models: the conditional inference tree (CITree) [12] (R package: party) and the recursive partitioning for survival trees (RPTree) [4,18] (R package: rpart). To conduct this experiment, I used the Kruskal–Wallis test, Dunn's multiple comparison test, and the Friedman test. A 0.05 significance level was applied for all the comparisons.

6.1. Synthetic data

The experiments were conducted using a synthetic data set simulated by the R package survsim [24]. The learning data consisted of 400 observations describing two covariates, x and y. As shown in Table 1, the data may be considered a sum of four data sets with different covariate distributions and Weibull distributions of survival and censoring times. The test data set contained 2000 uncensored samples.

Table 2
Results of synthetic data set obtained for different values of parameters η and ζ; χ^2 denotes the value of the χ^2 statistics in the root node, size denotes the number of tree nodes, MAE denotes the mean ± standard deviation of the mean absolute error, and IBS denotes the mean ± standard deviation of the integrated Brier score; each experiment was repeated 20 times.

ζ	χ^2	$\eta = min(0.3;1-\zeta)$			$\eta = min(0.7;1-\zeta)$		
		Size	MAE	IBS	Size	MAE	IBS
0.3	198.2	13	27.53 ± 0.86	0.089 ± 0.003	9	27.66 ± 0.63	0.089 ± 0.002
0.5	198.49	11	26.94 ± 0.84	0.089 ± 0.003	11	27.17 ± 1.07	0.09 ± 0.004
0.7	181.34	11	27.7 ± 0.86	0.094 ± 0.004	8	27.84 ± 0.9	0.094 ± 0.005
0.9	169.38	11	27.4 ± 1.25	0.092 ± 0.005	11	27.66 ± 1.13	0.093 ± 0.005

Table 3
Results of the synthetic data set obtained for the conditional inference tree (CITree) and the recursive partitioning for survival trees (RPTree); χ^2 denotes the value of the χ^2 statistics in the root node, Size denotes the number of tree nodes, MAE denotes the mean ± standard deviation of the mean absolute error, and IBS denotes the mean ± standard deviation of the integrated Brier score; each experiment was repeated 20 times.

	χ^2	Size	MAE	IBS
CITree	97.4	19	27.26 ± 0.0	0.048 ± 0.0004
RPTree	147	16	26.62 ± 0.9	0.13 ± 0.001

Table 2 shows the results of the synthetic data set using the dipolar survival tree. We tested the influence of the ζ and η parameters on the prediction ability of the tool, expressed as the MAE and the IBS. The measures were presented as the mean values and standard deviations calculated over 20 runs of the algorithm. The tree size was calculated as the median value.

Taking into account the MAE, the best result (26.94 ± 0.86) was for $\zeta = 0.5$ and $\eta = 0.3$ with the number of tree nodes equal to 11. For the IBS, the best results were for $\zeta = 0.3$ or 0.5, while the η values did not influence the results significantly. A similar conclusion may be drawn from the analysis of the χ^2 statistics in the root nodes. The greatest value was 198.49 (i.e., the best division) for $\zeta = 0.5$.

The results of the dipolar survival tree were compared with those of the two other tree-based methods: the CITree and the RPTree. All the experiments were repeated 20 times, and, in each run, the final pruned RPTree was selected by cross-validation. The outcomes, given in Table 3, are presented as the mean and standard deviations of the IBS and MAE.

Since the test data set does not contain censored observations, we used the MAE as a prediction ability measure. The Kruskal–Wallis test gave the statistically significant differences of the MAE between the tools ($p < 0.0001$). Dunn's multiple comparison test did not indicate any statistically significant differences between the DSTree and the CITree ($p > 0.05$). The comparison of the IBS gave similar results, except that Dunn's test gave the statistically significant differences ($p < 0.05$) for all the comparisons. Box-and-whisker plots of the MAE and the IBS are presented in Fig. 5.

Table 4
Mixed dipoles divided by the internal nodes of the tree in Fig. 6a; # denotes the number of mixed dipoles and divided mixed dipoles, % denotes the percent of the mixed dipoles and divided mixed dipoles relative to the whole data set, and % in the node denotes the percent of divided mixed dipoles relative to the subset of the data of a given node.

Node index	Mixed dipoles		Divided dipoles		
	#	%	#	% in node	%
0	37,339	100	25,212	67.5	67.5
1	6332	17	3585	56.6	9.6
2	1488	4	857	57.6	2.3
6	5795	15.5	3112	53.7	8.3
8	1154	3.1	861	74.6	2.3

The value of the χ^2 statistics indicates the quality of a split: the greater value, the better split and the bigger difference between two groups of patients. The χ^2 statistics in the root node obtained for the DSTree ranged from 169.38–198.49 (Table 2), for the RPTree was equal to 147, and for the CITree – 97.4 (Table 3). Therefore, in the whole synthetic data set, the minimization of the dipolar criterion function causes really good separation of observations with different survival times.

Fig. 6 presents the survival tree obtained for the synthetic data set with the parameters $\eta = 0.3$ and $\zeta = 0.5$. Each internal node (Fig. 6a) is described by the number of cases, the number of censored observations, the median survival time, the χ^2 statistics, and the split hyperplane equation. The tree includes five internal nodes that divide the feature space into six areas (Fig. 6b) represented by the terminal nodes. Each area is characterized by a median survival time and the Kaplan–Meier survival function, as shown in Fig. 6c. The log-rank test used to compare the obtained survival functions gives $p < 0.0001$, which indicates statistically significant differences between them. The best prediction is for cases from terminal node 4, for which the median survival time is equal to 183.43, the worst prediction is for terminal node 10 with the median survival time equal to 9.08.

More detailed descriptions of mixed and pure dipoles divided by internal nodes are given in Tables 4 and 5. The percentage of divided dipoles decreases with the level of the tree. The root node (level 0)

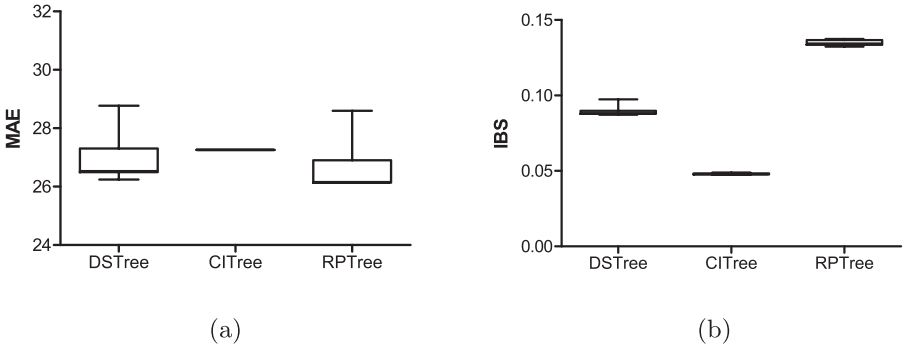


Fig. 5. Box-and-whisker plots of (a) the mean absolute error (MAE) and (b) the integrated Brier score (IBS) for three types of survival trees: the dipolar survival tree (DSTree), the conditional inference tree (CITree), and recursive partitioning for survival trees (RPTree).

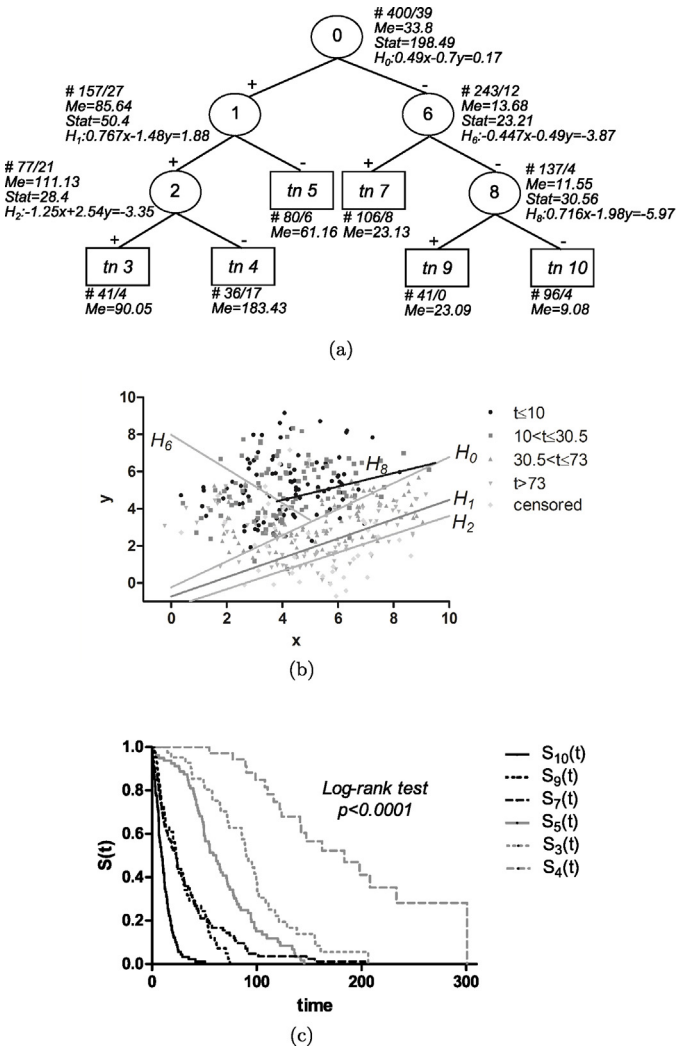


Fig. 6. Results of the synthetic data set ($\eta = 0.3$ and $\zeta = 0.5$): (a) dipolar survival tree; (b) division of the feature space; (c) Kaplan–Meier survival functions for distinguished areas (terminal nodes).

Table 5
Pure dipoles divided by the internal nodes of the tree in Fig. 6a; # denotes the number of pure dipoles and divided pure dipoles, % denotes the percent of pure dipoles and divided pure dipoles relative to the whole data set, and % in the node denotes the percent of divided pure dipoles relative to the subset of the data of a given node.

Node index	Pure dipoles		Divided dipoles		
	#	%	#	% in node	%
0	21,219	100	4239	20	20
1	1948	9.2	810	41.6	3.8
2	309	1.5	124	40.1	0.6
6	15,032	70.8	6909	46	32.6
8	6055	28.5	2067	34.1	9.7

splits 67.5% of mixed and only 20% of pure dipoles, while the nodes of the 2nd level (no. 2 and 8) divide 2.3% and 0.6–9.0% of mixed and pure dipoles, respectively. The percentage of the separated mixed dipoles relative to the dipoles that reached the node is 53.7–74.6%. As expected, the percentage of divided pure dipoles is less than that of the mixed dipoles; the in node percent is in the range of 20–46%. The tree, as a whole, divided 90.1% mixed dipoles and 66.1% pure dipoles.

Table 6
Description of data sets; #obs denotes the number of observations, %cens denotes the percent of censored cases, and #attr denotes the number of attributes.

Data set	#obs	%cens	#attr	Description
VALung [2]	137	6.5	6	Lung cancer
melanoma [25]	205	72.2	4	Malignant melanoma
pbc [26]	418	61.5	8	Primary biliary cirrhosis of the liver
nwtco [27]	668	87.3	5	National Wilms Tumor Study
bfeed [28,29]	927	3.8	8	Weaning of breast-fed newborns
kidtran [28,29]	863	83.8	3	Kidney transplant
larynx [28,29]	90	44.4	3	Laryngeal cancer

Table 7
Integrated Brier score of the three types of survival trees: the dipolar survival tree (DSTree), recursive partitioning for survival trees (RPTree), and the conditional inference tree (CITree); Size denotes the number of nodes and IBS denotes the mean ± standard deviation of the integrated Brier score.

Data set	DSTree		RPTree		CITree	
	Size	IBS	Size	IBS	Size	IBS
VALung	5	0.074 ± 0.003	15	0.088 ± 0.016	5	0.076 ± 0.008
melanoma	3	0.157 ± 0.007	15	0.117 ± 0.01	3	0.152 ± 0.003
pbc	5	0.14 ± 0.0027	27	0.068 ± 0.012	13	0.15 ± 0.006
nwtco	5	0.107 ± 0.002	7	0.071 ± 0.014	7	0.1 ± 0.001
bfeed	5	0.046 ± 0.0002	4	0.198 ± 0.007	5	0.074 ± 0.004
kidtran	5	0.13 ± 0.0004	5	0.106 ± 0.006	3	0.12 ± 0.0009
larynx	3	0.2 ± 0.009	5	0.155 ± 0.006	3	0.21 ± 0.005

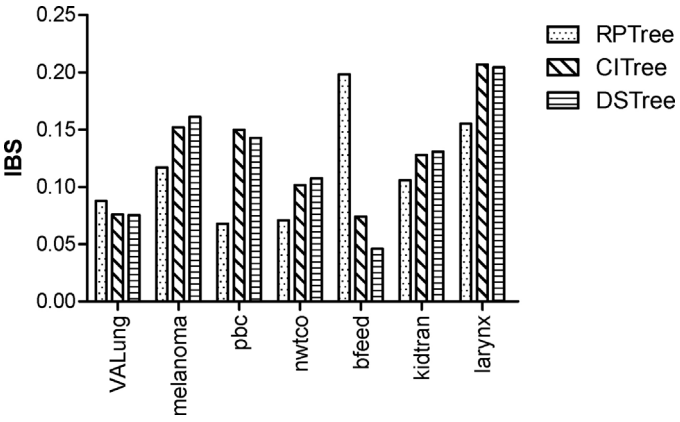


Fig. 7. Mean IBS values obtained for three types of survival trees: the dipolar survival tree (DSTree), the conditional inference tree (CITree), and recursive partitioning for survival trees (RPTree).

6.2. Real data sets

I used seven real data sets to compare the predictive ability of the dipolar survival tree with two other methods. Taking into account the results from previous experiments, the parameters ζ and η were fixed at 0.5 and 0.3, respectively. The data sets (see Table 6) have a diversified number of observations (90–927) with different percentages of censored cases (3.8–87.3%) and dimensions of feature vectors (3–8).

In Table 7, we can see the results of the experiments. The predictive ability of each method was described by the mean and standard deviations of IBS, while the tree sizes were calculated by the median value determined over 20 runs of the 10-fold cross-validation. The Friedman test was used to compare the methods' performance. It showed no statistically significant differences between the predictive ability of the models ($p = 0.486$) or the tree sizes ($p = 0.11$).

Although the statistical analysis did not indicate statistically significant differences between the methods, a more detailed analysis of the results may show some features of the DSTree. The size of the trees obtained using the dipolar criterion function is always not

greater than 5; only in the *bfeed* and *kidtran* data sets, the number of nodes in the DSTree is not the least. For the *VAlung* and *bfeed* data sets, the method outperformed the other two approaches, obtaining IBS values equal to 0.074 and 0.046, respectively. A graphical representation of the IBS of the three survival tree induction methods is given in Fig. 7.

7. Conclusions

In this paper, I developed an oblique survival tree (DSTree) induction algorithm that exploits the piecewise-linear criterion function as a splitting criterion. An analysis of the synthetic data set shows that the χ^2 statistics associated with the split in the root node of the DSTree is the greatest among all the compared methods, which indicates a very good separation of observations with different survival experiences.

Based on seven real data sets, two survival tree features (i.e., predictive ability and tree size) were compared with two already existing univariate tree models. Although the predictive ability of the dipolar tree is comparable to that of other tree induction algorithms, there are some data sets for which the division of the feature space made by the oblique trees is more advantageous. Taking into account the number of nodes, in the majority of cases, the DSTree is one of the smallest.

The experiments on real data sets were conducted with constant values of the algorithm parameters of dipole prices and the borders of time differences for dipole formation (i.e., η and ζ, respectively). The quality of the dipolar survival tree may be improved by adjusting these parameters to a given problem.

One of the main advantages of this method is its direct use of censored observations. Incomplete information from such data is incorporated at the stage of dipole creation and then utilized during the minimization of the dipolar function. Unfortunately, it is unlikely that all the censored observations will be included. If the data contains censored cases with smaller survival times compared to those of the uncensored observations, they will not form any dipoles and hence, it will not influence the positions of the hyperplanes in the internal nodes.

Dipole formation, and therefore the dipolar criterion function, has a strong relationship with the ranked measures of predictive ability. Here, only the evaluable pairs of the feature vectors (mixed or pure dipoles) are included in the learning process. Based on the percentage of dipoles divided, we can evaluate the quality of the split. This may be a sufficient alternative way of choosing the final survival tree instead of the split-complexity method presented in this paper.

The use of piece-wise linear criterion functions can be easily extended to other types of predictors, such as decision or regression trees. The only change required would be defining the dipole formation rules. In regression trees, the target value is a continuous variable without censoring, so the rules are similar to those of survival trees; for decision trees, the mixed dipoles should be created between feature vectors from different classes, while cases from the same class should form pure dipoles.

Acknowledgements

This work was supported by grant S/WI/2/2013 from Bialystok University of Technology founded by the Ministry of Science and Higher Education.

References

[1] Cox D. Regression models and life tables (with discussion). J R Stat Soc B 1972;34:187–220.
[2] Kalbfleisch J, Prentice R. The statistical analysis of failure time data. New York: John Wiley & Sons; 1980.
[3] LeBlanc M, Crowley J. A review of tree-based prognostic models, vol. 75 of Cancer treatment and research. US: Springer; 1995. p. 113–24.
[4] Breiman L, Friedman J, Olshen R, Stone C. Classification and regression trees. Belmont, CA: Wadsworth; 1984.
[5] Gordon L, Olshen R. Tree-structured survival analysis. Cancer Treat Rep 1985;69(10):1065–9.
[6] Davis R, Anderson J. Exponential survival trees. Stat Med 1989;8:947–61.
[7] LeBlanc M, Crowley J. Relative risk trees for censored survival data. Biometrics 1992;48:411–25.
[8] Therneau T, Grambsch P, Fleming T. Martingale-based residuals for survival models. Biometrika 1990;77(1):147–60, http://dx.doi.org/10.1093/biomet/77.1.147.
[9] 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 1986;4(3):185–204.
[10] Segal M. Regression trees for censored data. Biometrics 1988;44:35–47.
[11] LeBlanc M, Crowley J. Survival trees by goodness of split. J Am Stat Assoc 1993;88(422):457–67.
[12] Hothorn T, Hornik K, Zeileis A. Unbiased recursive partitioning: a conditional inference framework. J Comput Graph Stat 2006;15(3):651–74.
[13] Radespiel-Tröger M, Rabenstein T, Schneider HT, Lausen B. Comparison of tree-based methods for prognostic stratification of survival data. Artif Intell Med 2003;28(3):323–41.
[14] Radespiel-Tröger M, Gefeller O, Rabenstein T, Hothorn T. Association between split selection instability and predictive error in survival trees. Methods Inf Med 2006;45(5):548–56.
[15] Bou-Hamad I, Larocque D, Ben-Ameur H. A review of survival trees. Stat Surv 2011;5:44–71.
[16] Kretowska M. Dipolar regression trees in survival analysis. Biocybern Biomed Eng 2004;24(3):25–33.
[17] Bobrowski L, Kretowska M, Kretowski M. Design of neural classifying networks by using dipolar criterions. In: Tadeusiewicz R, Rutkowski L, Chojcan J, editors. Proceedings of the Third Conference on Neural Networks and Their Applications, Polish Society of Neural Networks. Poland: Kule; 1997. p. 689–94.
[18] Therneau T, Atkinson E. An introduction to recursive partitioning using the RPART routines. Mayo Foundation; 2015 http://CRAN.R-project.org/package=rpart.
[19] Kaplan E, Meier P. Nonparametric estimation from incomplete observations. J Am Stat Assoc 1958;53:457–81.
[20] Bobrowski L, Niemiro W. A method of synthesis of linear discriminant function in the case of nonseparability. Pattern Recognit 1984;17:205–10.
[21] Graf E, Schmoor C, Sauerbrei W, Schumacher M. Assessment and comparison of prognostic classification schemes for survival data. Stat Med 1999;18:2529–45.
[22] Harrell F, Califf R, Pryor D, Lee K, Rosati R. Evaluating the yield of medical tests. J Med Assoc 1982;247:2543–6.
[23] Korn EL, Simon R. Measures of explained variation for survival data. Stat Med 1990;9:487–503.
[24] Morina D, Navarro A. The R package survsim for the simulation of simple and complex survival data. J Stat Softw 2014;59(2):1–20.
[25] Andersen P, Borgan O, Gill R. Statistical models based on counting processes. New York: Springer; 1995.
[26] Fleming T, Harrington D. Counting processes and survival analysis. John Wiley & Sons; 1991.
[27] Therneau TM. survival: survival analysis, R package version 2.39; 2016 http://CRAN.R-project.org/package=survival.
[28] Klein J, Moeschberger M. Survival analysis, techniques for censored and truncated data. Springer; 1997.
[29] Klein J, Moeschberger M, Yan J. KMSurv: data sets from Klein and Moeschberger (1997), R package version 0.1-5; 2012 http://CRAN.R-project.org/package=KMSurv.