Censoring Weighted Separate-and-Conquer Rule Induction from Survival Data

Ł. Wróbel1,2; M. Sikora1,2

1Institute of Computer Science, Silesian University of Technology, Gliwice, Poland; 2Institute of Innovative Technologies EMAG, Katowice, Poland

Keywords
Survival prediction, rule induction, rule quality measures

Summary
Objectives: Rule induction is one of the major methods of machine learning. Rule-based models can be easily read and interpreted by humans, that makes them particularly useful in survival studies as they can help clinicians to better understand analysed data and make informed decisions about patient treatment. Although of such usefulness, there is still a little research on rule learning in survival analysis. In this paper we take a step towards rule-based analysis of survival data.

Methods: We investigate so-called covering or separate-and-conquer method of rule induction in combination with a weighting scheme for handling censored observations. We also focus on rule quality measures being one of the key elements differentiating particular implementations of separate-and-conquer rule induction algorithms. We examine 15 rule quality measures guiding rule induction process and reflecting a wide range of different rule learning heuristics.

Results: The algorithm is extensively tested on a collection of 20 real survival datasets and compared with the state-of-the-art survival trees and random survival forests algorithms. Most of the rule quality measures outperform Kaplan-Meier estimate and perform at least equally well as tree-based algorithms.

Conclusions: Separate-and-conquer rule induction in combination with weighting scheme is an effective technique for building rule-based models of survival data which, according to predictive accuracy, are competitive with tree-based algorithms.

1. Introduction

Survival analysis is widely used in biomedical and health studies to analyse time until the occurrence of an event such as death, disease incidence, relapse or recovery. Among a variety of different survival analysis methods, techniques for modelling the impact of covariates on survival time are of particular importance. The most popular such modelling methods are parametric and semi-parametric statistical techniques like the Cox proportional hazards regression [1] and its extensions. However, restrictive assumptions of these methods and difficulty in modelling nonlinear interactions between covariates are one of the motivations for developing new methods of survival analysis based on machine learning techniques. In this paper we investigate a nonparametric rule-based approach to modelling survival data.

Rule induction is one of the most frequently used methods of machine learning with numerous successful applications in a wide range of predictive and descriptive data mining tasks [2]. The comprehensibility of rule-based models makes rule induction a powerful tool for the analysis of biomedical and healthcare data. Researchers can explore high-dimensional interactions between covariates and find factors influencing the subject of analysis in the form of easily to understand logical rules. Understandable models can also be further investigated by domain experts in order to verify the results of analysis and share insights with expert community. Thus, rule induction methods can provide useful information for decision making as well as be the basis for further research.

Although great usefulness of rule-based systems in biomedical and healthcare sciences, there is still a little research on rule learning in survival analysis (cf. Section 1.1). Naturally, in the case of absence of censored observations the standard rule-based regression [3–5] techniques can be applied. However, as some study objects may not experience the event before the study ends, some may be lost to follow up during the study period or withdrawn from the study, the presence of censored observations is a key problem in survival analysis. By deleting such observations one would bias the model towards event outcome, whereas treating them as event-free would bias the model towards the event-free response. In this paper we investigate rule induction algorithm in combination with the weighting scheme which allows us to balance model between event and event-free outcome. The applied weighting method handles censored data by transformation of the problem of survival into...
two-class classification task with weighted observations.

As the basis of our rule induction method we selected separate-and-conquer (known also as covering) strategy [2, 6, 7] which is one of the most common ones employed in algorithms for induction of classification rules. Although considerable importance and wide research on separate-and-conquer rule learning applied for solving classification problems, the covering algorithms are still not well studied in the field of survival analysis. In this paper we present presumably the first such a comprehensive study of the separate-and-conquer rule learning from survival data.

1.1 Related Work

Methods of survival analysis are mainly used in medical studies. Although rule-based algorithms are often applied in medical research, there is a relatively small number of papers concerning application of rule induction to survival analysis.

Pattaraintakorn and Cercone [8] describe the rough sets-based intelligent system for survival analysis. The model construction part of this system uses a decision rule induction algorithm for identification of the main factors affecting survival time of patients. The survival time is considered as a discrete variable with predefined values (e.g. survival time between 56 and 73 months) dividing an entire dataset into separate decision classes. The rough set-based approach to survival analysis is also the subject of the paper [9]. For each observation in the analyzed dataset there is calculated prognostic index (PI) based on Cox’s proportional hazard model. A range of PI values is divided into three intervals, thereby creating separate groups differing in survival rate, and the rules are induced for classes created in such a way.

Sikora et al. [10] applied rule induction algorithm to the analysis of patients after bone marrow transplantation. The set of patients is divided into three groups: the patients for whom at least 5 years have passed since the transplantation (the class alive), the patients who died within 5 years after transplantation (the class dead), and the patients who are still alive but their survival time is less than 5 years (the class alive-5). Rules are generated for dataset containing alive and dead classes, whereas the alive-5 is used for the post-processing of obtained rules.

Kronek and Reddy [11] proposed the extension of Logical Analysis of Data (LAD) [12, 13] for survival analysis. The LAD algorithm is a combinatorial approach to rule induction. It was originally developed for the analysis of data containing binary attributes, therefore the data pre-processing by discretization and binarization methods is usually required.

Liu et al. [14] adapted patient rule induction method to the analysis of survival data. The method uses so-called bump hunting which creates rules by searching regions in covariates space with a high average value of the target variable. To deal with censoring, the authors use deviance residuals [15, 16] as the outcome variable. The idea of residual-based approach to censored outcome is derived from survival trees [15, 16].

Survival trees are an adaptation of classification and regression trees [17] to the problem of survival. In comparison with rule-based techniques, tree-based methods received much more attention in survival analysis [18]. On the other hand, a tree can be easily represented in the form of a set of rules where each path from the root to the leaf of the tree corresponds to one rule, thus it can be considered as a special case of the rule-based model. The key idea of the application of tree-based techniques to survival data lies in the splitting criterion [19]. The most popular approach, next to residual-based one, is to select a split which maximizes the difference between survival distributions of child nodes. It is often achieved by the maximization of the log-rank test statistics [20–22].

Wróbel [23] proposed to use a survival tree for induction of an ordered set of rules (decision list) from survival data. The core idea is to learn survival tree, extract the best rule from it, remove observations which are covered by the rule and recursively repeat such a procedure for remaining observations. This idea simply follows the approach used by the PART [24] and MSRules [3] algorithms for learning classification and regression rules respectively.

It should be noted that the aforementioned studies primarily concern an application of rule-based survival analysis to usually one, particular dataset. Pattaraintakorn and Cercone [8] mainly focus on geriatric data of Canadian patients. Bazan et al. [9] analyze data of patients with various kinds of the head and neck cancer cases, Sikora et al. [10] study the effects of bone marrow transplantation, Liu et al. [14] perform an analysis of kidney cancer tissue microarray data. Kronek and Reddy [11] propose a more general approach, however they verify the algorithm for only two real-life datasets. The exception is the work [23] where survival tree-based algorithm for decision list induction is tested on 15 various survival datasets.

1.2 Objectives and Outline

The main goal of this paper is to give a detailed description of separate-and-conquer approach to rule learning designed for the analysis of survival data (Section 2) and verify it on variety of survival problems (Section 3). In comparison with aforementioned related work, we propose a more general solution rather than the case-study one. Moreover, in contrast to [8, 9, 11], the presented method does not require data pre-processing with the use of discretization methods. It is particularly important for the quality of survival analysis because discretization can cause the loss of information, and the final performance of the model may strongly depend on a selected method of discretization.

The special attention is paid to rule quality measures which are one of the key elements of rule induction algorithms. Numerous empirical studies [25–29] show that rule quality measure guiding rule learning process is of great importance for predictive and descriptive abilities of the output set of rules. At present, in literature there has been proposed over 50 measures [25–27, 29–32]. As most of them are developed for classification purposes, it is not known how they would behave in survival analysis where the accuracy of predictions is measured in different way. Thus, in the experimental part of this study (Section 3) we examine the most popular quality...
measures in order to determine the most promising ones for examined algorithm. The performance of our rule-based framework for survival analysis is verified on a collection of 20 survival datasets describing a wide variety of real-life problems. We compare our solution with the state-of-art survival trees and random survival forests algorithms. The final remarks and direction of further work are provided in Section 4.

2. Methods

We denote survival data as D(A, T, δ). The D = {o1, ..., o|D|} is a set of cardinality |D| of observations. Each observation o_i is described by a triple (A, T, δ), where A = {A_1, ..., A_{|A|}} is a set of covariates (attributes), T is the observation time and δ is the censoring status. In this paper we assume, the most common one, right-censored model of data. Thus, for i-th observation the time T_i is the minimum of real survival time and right censoring time. The status δ_i = 1 if an event occurred and δ_i = 0 if i-th observation is right-censored. Each observation o_i can be represented as a vector o_i = (a_{i1}, ..., a_{i|A|}, T_i, δ_i), where a_{ij} is the value of i-th observation for the attribute A_j. The attributes can be of numerical as well as nominal type.

The output of our rule induction algorithm is a set containing rules in the following form:

IF c_1 AND c_2 AND ... THEN Š(T|A)

The premise of the rule consists of a conjunction of conditions c_j (j = 1, 2, ...). The condition c_j is an expression of the form A_j = a_j for nominal attributes and A_j < a_j or A_j ≥ a_j for numerical ones (A_j ∈ A and a_j is a value from the domain of the attribute A_j). The conclusion of the rule is an estimate Š(T|A) of the survival function calculated on the basis of observations covered by the rule, that is, satisfying all conditions of the premise. In this paper we estimate survival function by the Kaplan Meier (KM) method [33] (cf. formula in Section 3.3). Sample rules are presented in Figure 5 and Figure 6 in the experimental part of this study together with the KM survival curves corresponding to these rules (Figure 7).

2.1 Handling Censored Observations

The idea behinds proposed rule-based algorithm is to build a model which is able to discriminate observations which experienced an event from the observations which are likely to be event-free. Naturally, the observations with an event are directly specified in data (δ_i = 1), whereas potentially event-free observations are censored. The word ‘potentially’ should be emphasized because in fact, according to collected data, it is not known whether censored observation would be event-free or not. On the other hand, in medical research it is common to consider patients who survived longer than the specified time (for example 5 years for cancer studies [34]) as successfully cured, that is, event-free. Thus, we regard censored observations both as events and non-events assuming that the longer censored time the higher probability of being event-free. On the basis of these assumptions, in order to deal with censoring, we apply the weighting scheme which transforms survival dataset D(A, T, δ) into classification one D'(A, C, W) with weighted observations. The C is a class attribute having two possible values ⊕ and ⊖ denoting positive and negative class respectively, and W is a weight attribute. If observation o_i = (a_{ij}, T_i, δ_i) of dataset D experienced an event (δ_i = 1) then such an observation is represented in dataset D' as o'_i = (a_{ij}, ⊕, 1), i.e. o'_i belongs to positive class and it has assigned the weight equal to 1. However, if observation o_i is censored then it is represented in D' by two observations: (a_{ij}, ⊖, w_i) and (a_{ij}, ⊕, 1 – w_i). The weight w_i is calculated as follows:

w_i = \frac{\hat{S}(T_{max})}{\hat{S}(T)}

where the Š(T) is a value of the KM survival function estimate of the entire training set at time T, and T_max is maximum time recorded for censored observations. The weight w_i is simply the conditional probability of being event-free at time T_max for positive class than the observations censored later. Such a weighting procedure for learning from censored data was introduced by Zupan et al. in the case-study on recurrence of prostate cancer [35]. It was also used by Štajduhar and Dalbelo-Bašić [36] for learning Bayesian networks from survival data.

2.2 Rule Induction

After the transformation of the survival dataset into classification one, the rules are induced in order to describe patterns characterizing positive ⊕ and negative ⊖ classes. Finding a minimal set of such rules is a computationally expensive, therefore most of the rule induction algorithms use some heuristics. One of the most common one, used also in our implementation, is so-called separate-and-conquer or covering strategy [2, 6, 7]. This strategy consists in learning a rule which covers some part of a training set. Next, the observations covered by the learned rule are removed and the rule learning process starts recursively for the remaining observations. The outline of separate-and-conquer rule learning for a single class can be summarized as can be seen in Figure 1.

At each iteration of the while loop, one rule is learned. Induction is terminated if all observations of target class are covered by learned rules. As the RuleInduction function generates rules for one class only, the algorithm has to be run separately for each class of input dataset in order to find patterns describing all observations. The main part of the algorithm is the LearnRule function which is realized as can be seen in Figure 2.

Learning a single rule is performed by hill-climbing algorithm divided into two phases: growing (lines 1–13) and pruning (lines 14–23). In the growing phase a rule with no conditions in premise is initialized (line 1) and at each iteration of the repeat-until loop one condition is added (line 12). In order to select the best one, each possible condition is temporarily added to the rule and such a rule is evaluated (line 8) according to the specified quality measure. The condition with the highest value of the evaluation is selected. The set of possible conditions (line 4) is created based on the
values taken by observations covered by the rule (initially empty rule covers all observations). In the case of nominal attributes, the condition can take the form of \( A \leq a \), and for the numerical attributes it can take one of two forms: \( A < a \) or \( A \geq a \). For the numerical attributes the value \( a \) is an arithmetic mean between two successive values taken by observations for attribute \( A \). In order to prevent induction of too specific rules, the candidate condition is considered only if rule with such a condition covers at least \( \text{minCov} \) number of observations from the target class (lines 5–7). It should be noted that the \( \text{LearnRule} \) function takes observations of target class which are not yet covered by rules generated so far, therefore \( \text{minCov} \) parameter is decreased when the number of the \( \text{classUncovered} \) observations is less than \( \text{minCov} \).

The rule growing is stopped if the rule is precise (that is, all observations covered by the rule belong to target class) or if there are no conditions which might be added to the rule. After the rule growing phase, the rule is pruned (lines 14–23), that is, at each iteration of the repeat-until loop, there is deleted a condition without which the rule has the highest improvement in the value of quality measure.

Such a rule learning procedure, which consists in the pruning of the rule just after it has been learned, is similar to Incremental Reduced Error Pruning (I-REP) method of Fürnkranz and Widmer [37]. However, in contrast to I-REP we use the same quality measure in both phases and the same set observations for rule growing and pruning. Thus, a key factor affecting descriptive and predictive performance of rules is the quality measure specified as the parameter of the algorithm.

### 2.3 Handling Missing Attribute Values

Incomplete data is an important problem in the analysis of many real-life datasets. There are many different methods for dealing with missing attribute values. The simplest method is to delete observations with missings. However, with increasing number of missing values the amount of data wasted by such a method grows, therefore usually more advanced techniques are used. One of the most popular methods is so-called imputation [38] which replaces missing values with estimated ones. However, imputed values are still not a real data and special attention should be paid to avoid unexpected values. Thus, our implementation employs a so-called ignored value strategy [39] and rules are built based only on known values of observations. It is simply performed by skipping missing values during search of possible conditions. If observation has a missing value for an attribute tested by the rule then such an observation is not covered by this rule. In contrast to imputation methods, this strategy does not require any additional computations and as was shown in [39] it performs similarly to more advanced methods for handling missing values.

### 2.4 Rule Evaluation

Evaluation of candidate rules is one of the key elements of classification rule learners. The aim of most rule quality measures is to guide rule induction in such a way that learned rule covers a large number of observations from the target class (let us denote this number as \( p \)) and a small number of observation form opposite classes (denoted as \( n \)) as possible. As the increase in value of \( p \) usually goes hand-in-hand with the increase in value of \( n \), the rule quality measure tries to find a trade-off between these two values, that is, between coverage and consistency.

Examined in this paper quality measures are presented in Table 1 (we describe their source a bit later). It should be noted that some measures take also the size of the target class (denoted as \( P \)) as well as the number of all observations from the opposite class (denoted as \( N \)) into account. Moreover, some of them are parameterized, that is, they require to specify the value of additional parameter (for example, \( m \) for the M-estimate). The detailed description of these parameters can be found, inter alia, in [2, 27].
The values $P$, $N$ as well as measure's parameter are constant during induction of rules for target class, therefore, the presented measures can be seen as functions of the $p$ and $n$, thus they solely depend on the rule being evaluated. In order to handle the weights of censored observations, the way of calculation of the values $p$, $n$, $P$, $N$ has to be slightly changed. Thus, the $p$, $n$, $P$, $N$ values are calculated as the sum of weights of corresponding observations, that is, the $p$ is the sum of weights of the observations from the target class which are covered by the rule, and $P$ is the sum of weights of all observations from the target class. The values $n$ and $N$ are calculated analogously.

At present, over 50 rule quality measures can be found in the literature [26, 27, 29–32]. In this paper we focus on measures presented in recent papers of Janssen and Fürnkranz [27] and Sikora and Wróbel [29] covering a quite wide range of different learning heuristics and reflecting various rule learning behaviours. Janssen and Fürnkranz [27] verified 10 rule quality measures known as Accuracy, Correlation, Cost, F-measure, Klösgen, Laplace, M-estimate, Precision, Relative Cost and WRA. Moreover, for parameterized measures they provide optimal values of the parameters, that is, $c = 0.437$ for Cost, $\beta = 0.5$ for F-measure, $\omega = 0.4323$ for Klösgen, $m = 22.466$ for M-estimate and $c_r = 0.342$ for Relative Cost (we also use these recommended values in our experiments). Sikora and Wróbel [29] identified a set of nine measures the most effective according to predictive and descriptive abilities of the induced rules. These measures are: $C_1$, $C_2$, Correlation, G-measure with parameter $g = 2$, Logical Sufficiency, Mutual Support, RSS, s-Bayesian Confirmation and Weighted Laplace. In the experimental part of this paper we investigate all of these measures. However, it should be noted that Correlation occurs in both [27] and [29], as well as some measures presented in [27] are equivalent to those in [29]. In brief, we consider two measures $q_1$ and $q_2$ as equivalent if the RuleInduction function invoked with $q_1$ parameter returns the same set of rules as the RuleInduction function with $q_2$ parameter. The equivalent pairs of measures are: Laplace and Weighted Laplace, Precision and Logical Sufficiency, WRA and RSS. Therefore, the final set of examined measures consists of 15 different ones (Table 1).

### 2.5 Rule-based Estimation of Survival Probability

The learned rule set, firstly, describes survival patterns characterizing the training data, and secondly, it can be applied for an estimation of the survival function of new observations based on the values taken by their covariates. The estimation is performed by the rules covering a given observation, thus, the three situations should be taken into account: 1) none of the induced rules cover the observation, 2) the observation is covered by exactly one rule and 3) there are many different rules covering the observation. Naturally, in the second case, the observation has assigned an estimate of the survival function of the only matching rule. In the first case, we assign the default estimate to the observation, that is, the estimate which is calculated based on all observations of the training set. In the third case, we calculate the final estimate as the average of survival estimates of all matching rules. In this case the survival probability at time $T_i$ of observation $o_i$ is calculated as follows:

$$ \hat{S}(T_i | A) = \frac{1}{n} \sum_{r=1}^{n} \hat{S}_r(T_i | A) $$

where $n$ is the number of rules covering $o_i$, and $\hat{S}_r(T_i | A)$ is the estimate pointed by $r$-th rule. The similar averaging-approach is used by Kronek and Reddy [11]. However, in contrast to [11] we do not take into ac-
count survival estimate of entire training set in the above sum. Our preliminary experiments showed that it leads to better predictive performance of the presented separate-and-conquer framework. In our case, the default estimate is only assigned to the observations which are not covered by any of the rules.

### 3. Experimental Study

#### 3.1 Experimental Setup

In the experimental part of this study we address the question of how censoring weighted rule induction approach performs on a wide variety of survival problems. Therefore, we identify which rule quality measures lead to the lowest prediction error and compare them with the results of other approaches. The rule induction algorithm is run independently for each measure from Table 1. We refer to such standalone instances of the rule induction algorithm by the name of measure guiding rule induction process. For all measures the \( \min\text{Cov} \) parameter of the algorithm is set to seven observations that corresponds to default value of the \( \min\text{bucket} \) parameter determining the minimum number of observations in a terminal node for the examined implementations of survival trees (Section 3.3).

#### 3.2 Datasets

The experiments were carried out on 20 publicly available survival datasets. Their characteristics are shown in Table 2. They differ in the number of observations and attributes as well as in the number of censored observations, some of them also have missing attribute values. Most of these datasets can be found in various packages of the R environment [40]. It should be noted that some of the analyzed data contain attributes which are not taken into account during model building (for example id-like attributes), thus the number of attributes reported in Table 2 concerns only variables which were used for model construction.

#### 3.3 Other Tested Approaches

The results of quality measure-driven rule induction are compared with the results achieved by other machine learning methods which are able to express survival data model in the comprehensible form. In this regard the most popular are survival trees (cf. Section 1.1). We tested two implementations of such approaches. Both of them are available in the R environment. The first one is the RPART algorithm [43] from \( \text{party} \) package. The CTREE builds a tree from survival data using a splitting criterion based on the log-rank statistic.

We also examined extensions of survival trees in the form of random forests [45] which nowadays are considered as one of the most powerful machine learning methods according to predictive accuracy. In our experimental study we use random survival forest implementation available in the R package \( \text{randomSurvivalForest} \) (function RSF) [46] and conditional inference forest (function CFOREST) [47] from the package \( \text{party} \). All algorithms are run in their default configurations except RSF algorithm for which the number of trees to grow in a forest is set to 500 (it is the default value for the CFOREST).

Moreover, we also provide the results of the Kaplan-Meier (KM) estimator calculated based on the entire training set. The KM estimator of survival function is given as:

\[
\hat{S}(t) = \prod_{i \in \mathcal{E}(t)} \left(1 - \frac{d_i}{r_i}\right)
\]

where \( d_i \) is the number of observations which experienced an event at time \( T_i \), and \( r_i \) is the number of subjects at risk, that is, which are still observable at time \( T_i \). The KM estimator ignores the influence of covariates and at a given time it assigns the same survival probability for all observa-

### Table 1 The rule quality measures used in the experimental part of this study

<table>
<thead>
<tr>
<th>Measure</th>
<th>Formula</th>
<th>Measure</th>
<th>Formula</th>
<th>Measure</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>( p - n )</td>
<td>F-measure</td>
<td>( \frac{(p^2 + 1)(p^2 + n)}{p^2(p + n)} )</td>
<td>Mutual Support</td>
<td>( \frac{p}{n + p} )</td>
</tr>
<tr>
<td>C1</td>
<td>( \frac{(np - p) - 2 + \text{Cohen}}{N(p + n)} )</td>
<td>G-measure</td>
<td>( \frac{p}{n + p + g} )</td>
<td>Precision</td>
<td>( \frac{p}{n + p} )</td>
</tr>
<tr>
<td>C2</td>
<td>( \frac{(np - p) + p + N}{np} )</td>
<td>Klösgen</td>
<td>( \frac{p + n + p}{n + p + N} )</td>
<td>Relative Cost</td>
<td>( \frac{p - (n + p)}{n} )</td>
</tr>
<tr>
<td>Correlation</td>
<td>( \frac{pN - p_n}{np\sqrt{p + n}(p + N - p)} )</td>
<td>Laplace</td>
<td>( \frac{p + 1}{p + n + 2} )</td>
<td>RSS</td>
<td>( \frac{p - n}{p} )</td>
</tr>
<tr>
<td>Cost</td>
<td>( cP - (1 - c)n )</td>
<td>M-estimate</td>
<td>( \frac{p + m}{p + n + m} )</td>
<td>s-Bayesian Confirmation</td>
<td>( \frac{p}{n + p} )</td>
</tr>
</tbody>
</table>
3.4 Evaluation

Each algorithm is tested with the use of 10-fold stratified cross-validation repeated 10 times on each dataset. The final performance achieved by the algorithm on particular dataset is calculated as an arithmetic mean from 10 train-and-test iterations. The stratification of survival data is performed according to the censoring status, that is, the proportion of events to censored observations in each fold is approximately equal to such proportion in the entire training set.

Algorithms are evaluated according to two criteria: size of the learned models and their predictive accuracy. The size is calculated as the number of rules for rule-based models, the number of leaves for trees and the number of trees for random forests. The accuracy of predictions is measured by the integrated Brier score [48, 49] which is recently one of the most popular criteria for the evaluation of survival models. The Brier score (BS) for survival data is expressed by the formula shown in Figure 3.

It calculates the squared difference between true event status at given time \( T^* \) and the predicted event status \( p(T^*) \) at that time over all \( n \) observations in the test set. The true event status for \( i \)-th observation is equal to 0 if an event occurred for this observation before or at the time \( T^* \), and it is equal to 1 if survival time \( T_i \) of the observation is greater than \( T^* \) (the \( I \) in the BS formula is the indicator function). The predicted event status is returned by learned model as a survival probability at a given time \( T^* \). The censoring is taken into account by weighting the squared differences by the inverse of the estimate \( G \) of the censoring survival function. The \( G \) estimate is simply calculated as the KM estimator based on training observations with censoring status set to \( (1 - \delta) \). The Brier score is a cost-type criterion i.e. the lower the value the more accurate the prediction. The integrated Brier score (IBS) summarizes the prediction error over all times in a test set and it is given by:

\[
IBS = \frac{1}{n \max(T_i)} \int_0^{\max(T_i)} BS(T^*)dT^*
\]

In our experimental study we compare algorithms on multiple datasets with the use of statistical tests recommend by Demšar [50]. For comparison of two classifiers according to the IBS on multiple datasets we use the Wilcoxon signed rank test and for comparison of all algorithms with each other we apply the Friedman test with a post-hoc Nemenyi test. The Friedman test ranks all considered algorithms for each dataset separately. The rank 1 is assigned to the algorithm with the lowest IBS, the rank 2 – to the second algorithm with the lowest IBS and so on. Finally, each algorithm has assigned an average rank as an arithmetic mean from 100 train-and-test iterations. The stratification of survival data is performed according to the censoring status set to \( (1 - \delta) \) based on training observations with censoring survival function. The IBS estimator is simply calculated as the KM estimator based on training observations with censoring status set to \( (1 - \delta) \).

### Table 2

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#obs</th>
<th>#attr</th>
<th>%miss</th>
<th>%cens</th>
<th>Study subject</th>
</tr>
</thead>
<tbody>
<tr>
<td>actg320 [41]</td>
<td>1151</td>
<td>11</td>
<td>0</td>
<td>91.66</td>
<td>HIV-infected patients</td>
</tr>
<tr>
<td>cancer [R]</td>
<td>228</td>
<td>7</td>
<td>4.14</td>
<td>27.63</td>
<td>advanced lung cancer patients</td>
</tr>
<tr>
<td>follic [R]</td>
<td>541</td>
<td>4</td>
<td>0</td>
<td>35.67</td>
<td>follicular cell lymphoma patients</td>
</tr>
<tr>
<td>GBSG2 [R]</td>
<td>686</td>
<td>8</td>
<td>0</td>
<td>56.41</td>
<td>node-positive breast cancer patients</td>
</tr>
<tr>
<td>halibut [42]</td>
<td>294</td>
<td>5</td>
<td>0</td>
<td>7.14</td>
<td>size regulation of Atlantic halibut</td>
</tr>
<tr>
<td>hd [R]</td>
<td>865</td>
<td>6</td>
<td>0</td>
<td>50.75</td>
<td>Hodgkin’s disease patients</td>
</tr>
<tr>
<td>lung [42]</td>
<td>1032</td>
<td>7</td>
<td>2.6</td>
<td>25.97</td>
<td>early detection of lung cancer</td>
</tr>
<tr>
<td>Melanoma [R]</td>
<td>205</td>
<td>7</td>
<td>0</td>
<td>65.37</td>
<td>malignant melanoma patients after radical operation</td>
</tr>
<tr>
<td>mgus [R]</td>
<td>241</td>
<td>9</td>
<td>19.59</td>
<td>23.65</td>
<td>patients with monoclonal gammopathy of undetermined significance</td>
</tr>
<tr>
<td>nursing [42]</td>
<td>1601</td>
<td>5</td>
<td>0</td>
<td>20.11</td>
<td>duration analysis of nursing home usage</td>
</tr>
<tr>
<td>pbc [R]</td>
<td>418</td>
<td>17</td>
<td>14.54</td>
<td>61.48</td>
<td>primary biliary cirrhosis of the liver</td>
</tr>
<tr>
<td>Rossi [R]</td>
<td>432</td>
<td>60</td>
<td>10.24</td>
<td>73.61</td>
<td>observation of convicts released from Maryland state prisons</td>
</tr>
<tr>
<td>std [R]</td>
<td>877</td>
<td>21</td>
<td>0</td>
<td>60.43</td>
<td>occurrence of sexually transmitted diseases</td>
</tr>
<tr>
<td>TRACE [R]</td>
<td>1878</td>
<td>6</td>
<td>0</td>
<td>48.35</td>
<td>survival of patients after myocardial infarction</td>
</tr>
<tr>
<td>uis [R]</td>
<td>575</td>
<td>13</td>
<td>0</td>
<td>19.3</td>
<td>drug abuse reduction</td>
</tr>
<tr>
<td>unemployed [R]</td>
<td>300</td>
<td>6</td>
<td>0</td>
<td>56.67</td>
<td>time spent in the unemployment spell</td>
</tr>
<tr>
<td>wcgs [R]</td>
<td>3154</td>
<td>10</td>
<td>0.04</td>
<td>91.85</td>
<td>occurrence of coronary heart disease</td>
</tr>
<tr>
<td>whas1 [41]</td>
<td>481</td>
<td>7</td>
<td>0</td>
<td>48.23</td>
<td>myocardial infarction patients (1st book edition)</td>
</tr>
<tr>
<td>whas500 [41]</td>
<td>500</td>
<td>13</td>
<td>0</td>
<td>57</td>
<td>myocardial infarction patients (2nd book edition)</td>
</tr>
<tr>
<td>zinc [R]</td>
<td>431</td>
<td>55</td>
<td>57.17</td>
<td>81.21</td>
<td>esophageal cancer</td>
</tr>
</tbody>
</table>

![Figure 3](image-url)

**Figure 3**: Brier score formula

\[
BS(T^*) = \frac{1}{n} \sum_{i=1}^{n} \left\{ \left[ (1 - p(T^*)) \right]^2 I(T_i \leq T^* \land \delta_i = 1) \cdot G^{-1}(T_i) + \left[ (1 - p(T^*)) \right]^2 I(T_i > T^*) \cdot G^{-1}(T_i) \right\}
\]
states that all algorithms perform equally in mean ranking. If the null-hypothesis is rejected, we can perform the Nemenyi post-hoc test for pairwise comparison of methods. Two algorithms are significantly different if their mean ranks differ by at least the critical difference (CD). The results of the Nemenyi test are often visualized in the form of so-called critical difference diagram [50] (Figure 4).

3.5 Results and Discussion

We evaluated 20 algorithms in total, namely 15 instances of rule induction algorithm (each one corresponds to the one quality measure from Table 1), two algorithms of survival trees, two implementations of random survival forests and finally the KM estimator which serves as our reference null model. The Friedman test rejects (with p-value much smaller than 0.05) null hypothesis that the algorithms perform equally well according to the IBS. The comparison of all algorithms against each other with post-hoc Nemenyi test at 0.05 significance level is summarized in Figure 4. The figure shows a critical difference diagram [50] with all algorithms ranked according to the IBS. The ranks are marked on the axis, but their exact values are also given beside the name of each algorithm. Moreover, in the parentheses, for each algorithm there is also given a median (rounded to the nearest integer) of the model size over all datasets. The algorithms which are not significantly different at the 0.05 level are connected. The lowest average IBS rank was obtained by the CFOREST implementation of random survival forest, but according to the Nemenyi test there is no difference between the best performing CFOREST and first eight rule quality measures. Interesting enough, only CFOREST, M-estimate, C2, G-measure, RSF, Klösgen and C1 seem to perform significantly better than the KM estimate. However, a large gap can be observed between the KM estimate and most of other approaches. As the Nemenyi test is known to be very conservative [51], it might be not powerful enough to show differences in this case. Similarly, the CFOREST seems to deviate from other approaches.

The detailed comparison of the predictive performance of the best ranked measure, i.e. M-estimate, with other methods is given in Table 3. The table presents IBS (rounded to four decimal digits) for each considered survival dataset. For each dataset we also mark with bold font the lowest IBS values among all single model-based approaches – we exclude from this comparison the ensemble algorithms (RSF and CFOREST columns) because it is generally known that they tend to have lower prediction error at the expense of much less comprehensible data representations, thus confronting them with single model-based methods would be unfair. In the last row we provide the p-values of the Wilcoxon signed rank test confronting M-estimate with other approaches. The sing < (>) next to the p-value indicates that the M-estimate tends to have smaller (larger) prediction error than the second algorithm. Additionally, in parentheses next to the value of IBS, for the M-estimate, RPART and CFOREST algorithms we present an average model size for each dataset.

In general, the Wilcoxon test confirms the insights of critical difference diagram from Figure 4. The only exception is the CFOREST which significantly outperforms the KM estimate. Nevertheless, according to the Wilcoxon test, the M-estimate is competitive with the RPART, CTREE and RSF algorithms and significantly outperforms the KM estimate. Moreover, the p-value of the Wilcoxon test comparing M-estimate with RPART is below 0.1, thus it indicates generally better predictive performance of M-estimate over the RPART.

The learned models also differ in size (Figure 4). The most complex are random forests, whereas the simplest are survival trees. The size of rule-based models depends on the quality measure used for rule induction. The most compact are rule sets generated by S, Mutual Support and F-measure, however according to the IBS criterion they are ranked worse than the
KM estimator. In general, it can be observed that the measures which generate larger rule sets tend to have a lower prediction error. As these measures favour more precise rules at the expense of their lower coverage (thus leading to more complex rule sets), it seems that the estimation of survival probability based on such rules is more accurate. On the other hand, the measures like Precision and Laplace are ranked on average three positions below the M-estimate and induce more rules at the same time, therefore they have a tendency for over-fitting.

Generally, the obtained rule-based models are more complex than survival trees. One of the reasons of larger rule sets may lie in the fact that the applied weighting scheme enlarges input training set by duplicating censored observations, thus affecting the final number of rules. Moreover, besides minbucket parameter, the RPART and CTREE use additional pre-pruning minsplit criterion which determines the minimum number of observations in the node that is required to perform a split (the default value of this parameter is set to 20 for both implementations). Furthermore, the CTREE considers splitting if adjusted p-value of the log-rank test of the split is below the specified significance level (default 0.05), if such a split is not found, the tree building is stopped. For some applications the aforementioned tree-based data models may be however too simple and mainly reflect well-known dependencies, thus in such cases rule-based representations can be more beneficial for discovery of new, previously unknown knowledge from data.

3.6 An Illustrative Example

The rule sets containing dozens of rules are nevertheless still quite easy to explore for potentially useful information. For example, a user can filter out the rules which do not satisfy the specified quality criteria or these which do not contain attributes interesting for a user. In order to demonstrate sample rule filtering procedure we will analyse the rule set generated by the M-estimate for German Breast Cancer Study Group 2 [52] data (denoted in Table 2 as GBSG2). The dataset describes patients with primary node positive breast cancer and it is also used, inter alia, in [11, 36, 44, 47, 48] to test different modelling techniques. Each observation is described by the following attributes: hormonal therapy (horTh), age, menopausal status (menostat), tumour size (tsize), tumour grade (tgrade), number of positive nodes (pnodes), progesterone receptor (progrec), estrogen receptor (estrec).

The rule set induced by the M-estimate for the whole GBSG2 data consists of 134 rules, therefore in order to summarize the knowledge expressed by these rules we applied the following heuristic method enabling us to choose the rules describing the most different groups of observations. First, we selected the best, according to the M-estimate, rule from rules generated for the positive class and separately from rules of the negative class, thereby creating an initial rule set containing two rules. Next, this rule set was successively expanded by adding such rules that cover the largest number of observations yet uncovered by the rules already included in the set. Such a procedure may bias towards too general rules (i.e. covering the largest number of observations), therefore we limited our search to these rules which cover at most 33% of the entire dataset that approximately corresponds to the size of the largest group determined by the CTREE algorithm for GBSG2 data. Moreover, in order to limit the size of the final rule set we stopped construction of the model after adding a sixth rule. The ‘6 rules’ parameter corresponds to the number of leaves in the RPART tree generated for GBSG2 set as well as the size of the LASD model of GBSG2 data reported in [11] by Kronke and Reddy. As the result of such a simple

<table>
<thead>
<tr>
<th>Dataset</th>
<th>M-estimate</th>
<th>KM</th>
<th>RPART</th>
<th>CTREE</th>
<th>RSF</th>
<th>CFOREST</th>
</tr>
</thead>
<tbody>
<tr>
<td>actg32</td>
<td>0.0581 (191)</td>
<td>0.0612</td>
<td>0.0599 (7)</td>
<td>0.0595 (4)</td>
<td>0.0580</td>
<td>0.0576</td>
</tr>
<tr>
<td>cancer</td>
<td>0.1540 (29)</td>
<td>0.1521</td>
<td>0.1709 (10)</td>
<td>0.1553 (3)</td>
<td>0.1557</td>
<td>0.1506</td>
</tr>
<tr>
<td>follic</td>
<td>0.1838 (75)</td>
<td>0.1925</td>
<td>0.1863 (5)</td>
<td>0.1891 (4)</td>
<td>0.1949</td>
<td>0.1876</td>
</tr>
<tr>
<td>GBSG2</td>
<td>0.1666 (120)</td>
<td>0.1879</td>
<td>0.1778 (6)</td>
<td>0.1796 (4)</td>
<td>0.1657</td>
<td>0.1652</td>
</tr>
<tr>
<td>halibut</td>
<td>0.0729 (13)</td>
<td>0.0766</td>
<td>0.0658 (12)</td>
<td>0.0652 (7)</td>
<td>0.0659</td>
<td>0.0563</td>
</tr>
<tr>
<td>hd</td>
<td>0.2171 (118)</td>
<td>0.2242</td>
<td>0.2102 (2)</td>
<td>0.2085 (3)</td>
<td>0.2153</td>
<td>0.2133</td>
</tr>
<tr>
<td>lung</td>
<td>0.1444 (54)</td>
<td>0.1615</td>
<td>0.1431 (4)</td>
<td>0.1430 (5)</td>
<td>0.1467</td>
<td>0.1411</td>
</tr>
<tr>
<td>Melanoma</td>
<td>0.1651 (31)</td>
<td>0.1832</td>
<td>0.1820 (10)</td>
<td>0.1818 (3)</td>
<td>0.1643</td>
<td>0.1603</td>
</tr>
<tr>
<td>mgus</td>
<td>0.1709 (27)</td>
<td>0.1844</td>
<td>0.1833 (10)</td>
<td>0.1701 (5)</td>
<td>0.1604</td>
<td>0.1568</td>
</tr>
<tr>
<td>nursing</td>
<td>0.1748 (141)</td>
<td>0.1750</td>
<td>0.1766 (2)</td>
<td>0.1756 (5)</td>
<td>0.1879</td>
<td>0.1819</td>
</tr>
<tr>
<td>pbc</td>
<td>0.1450 (75)</td>
<td>0.1882</td>
<td>0.1713 (14)</td>
<td>0.1547 (6)</td>
<td>0.1359</td>
<td>0.1321</td>
</tr>
<tr>
<td>Rossi</td>
<td>0.1475 (21)</td>
<td>0.1233</td>
<td>0.1582 (10)</td>
<td>0.1589 (2)</td>
<td>0.1555</td>
<td>0.1343</td>
</tr>
<tr>
<td>std</td>
<td>0.2187 (161)</td>
<td>0.2083</td>
<td>0.2239 (3)</td>
<td>0.2219 (4)</td>
<td>0.2169</td>
<td>0.2141</td>
</tr>
<tr>
<td>TRACE</td>
<td>0.1802 (278)</td>
<td>0.2192</td>
<td>0.1801 (6)</td>
<td>0.1754 (14)</td>
<td>0.1706</td>
<td>0.1687</td>
</tr>
<tr>
<td>uis</td>
<td>0.1576 (69)</td>
<td>0.1643</td>
<td>0.1541 (12)</td>
<td>0.1537 (5)</td>
<td>0.1530</td>
<td>0.1461</td>
</tr>
<tr>
<td>unemployed</td>
<td>0.1941 (39)</td>
<td>0.2087</td>
<td>0.1756 (5)</td>
<td>0.1630 (3)</td>
<td>0.1975</td>
<td>0.1705</td>
</tr>
<tr>
<td>wcgs</td>
<td>0.0423 (664)</td>
<td>0.0439</td>
<td>0.0439 (6)</td>
<td>0.0436 (10)</td>
<td>0.0413</td>
<td>0.0419</td>
</tr>
<tr>
<td>whas1</td>
<td>0.2015 (64)</td>
<td>0.2318</td>
<td>0.1969 (7)</td>
<td>0.1942 (6)</td>
<td>0.1915</td>
<td>0.1923</td>
</tr>
<tr>
<td>whas500</td>
<td>0.1840 (83)</td>
<td>0.2242</td>
<td>0.1909 (11)</td>
<td>0.1890 (6)</td>
<td>0.1755</td>
<td>0.1755</td>
</tr>
<tr>
<td>zinc</td>
<td>0.0955 (46)</td>
<td>0.1095</td>
<td>0.1110 (9)</td>
<td>0.1044 (2)</td>
<td>0.0992</td>
<td>0.1011</td>
</tr>
</tbody>
</table>

Wilcoxon for M-estimate, Kaplan-Meier (KM) estimator, RPART and CTREE survival trees, RSF and CFOREST random survival forests on the 20 survival datasets. The last row shows the p-value of the Wilcoxon signed rank test confronting M-estimate with other approaches. Additionally, for M-estimate, RPART and CTREE an average model size is provided in parentheses next to the value of IBS.
(R1) \( \text{pnodes} \geq 9.5 \textbf{ and } \text{prorec} < 25 \textbf{ and } \text{estrec} \geq 0.5 \)

(R2) \( \text{pnodes} \in [3; 43.5] \textbf{ and } \text{age} \geq 49.5 \textbf{ and } \text{menostat} = \text{Post} \textbf{ and } \text{prorec} < 2380 \)

(R3) \( \text{horTh} = \text{no} \textbf{ and } \text{tsize} \in (15; 27.5) \textbf{ and } \text{prorec} < 980 \textbf{ and } \text{estrec} < 746.5 \textbf{ and } \text{age} < 69.5 \)

Figure 5  Rules obtained for the positive class of GBSG2 set

(R4) \( \text{pnodes} < 9 \textbf{ and } \text{prorec} \in [3; 279.5] \textbf{ and } \text{estrec} < 59.5 \textbf{ and } \text{age} \geq 34 \)

(R5) \( \text{pnodes} < 2.5 \textbf{ and } \text{estrec} \in [6; 574] \textbf{ and } \text{age} \geq 45 \textbf{ and } \text{tsize} < 46.5 \)

(R6) \( \text{pnodes} < 2.5 \textbf{ and } \text{horTh} = \text{yes} \textbf{ and } \text{prorec} \geq 22 \textbf{ and } \text{estrec} < 749 \)

Figure 6  Rules obtained for the negative class of GBSG2 set

Figure 7  The KM survival curves for observations covered by the rules R1–R6 induced from GBSG2 data. The dashed curve represents the KM estimate of the whole GBSG2 set.

rule filtering method we obtained model describing about 80% of all observations and containing the rules for the positive class (Figure 5) and the rules for the negative class (Figure 6).

The KM survival curves for observations covered by the R1-R6 rules are presented in Figure 7. Additionally, the graph shows dashed curve named as default representing the KM estimate for the whole GBSG2 dataset. The significant difference can be observed between the curves determined by the rules R1-R3 of the positive class and the curves of observations covered by the rules R4-R6 of the negative class. The R4-R6 curves are placed above the default KM estimate, and the observations covered by R4-R6 are characterized by a generally higher survival rate than the observations covered by R1-R3 rules.

4. Conclusions and Future Work

In this paper we addressed the issue of rule induction from survival data. Although rule induction is one of the oldest and popular methods of machine learning, there is still a little research on this topic in survival analysis. Since a separate-and-conquer strategy is widely used approach to rule learning, we decided to examine it in more detail in combination with the weighting scheme for handling censored observations. As the result, we have presented presumably the first such an extensive study on a separate-and-conquer rule learning applied to survival analysis. The described method covers all important aspects of rule-based analysis designed for survival data.

In general, our experimental study has shown that according to the predictive accuracy the rule induction outperforms the Kaplan-Meier estimator and it is a competitive technique with the state-of-art machine learning methods like survival trees and random survival forests. As the covering strategy is typical for most rule induction algorithms, we have focused on rule quality measures being one of the key factors differentiating particular implementations of the rule induction algorithms. We have examined 15 well-known measures which reflect various rule learning behaviours. Summarizing the results achieved by a particular rule learning heuristics, we believe that in most survival data mining tasks it would be enough to experiment with the M-estimate \( m = 22.466 \), C2, Klösgen \( \omega = 0.4323 \), Correlation or RSS measures. According to our experimental investigation, the M-estimate and C2 are characterized by the lowest prediction error at the cost of larger rule sets, the Klösgen measure generates less rules, but it also has lower accuracy of predictions, whereas Correlation and RSS induce compact rule sets which usually perform better than the Kaplan-Meier estimator.

The primary result of this research is a thorough study of separate-and-conquer rule learning designed for the analysis of survival data. We have focused mainly on detailed description of the rule induction algorithm adapted to survival domain and its extensive empirical evaluation on a large
Acknowledgments

The first author was a scholarship holder of project "SWIFT" POKL-08.02.01-24-005/10 co-financed by the European Union within the European Social Fund. The research of the second author was supported by the National Science Centre based on the decision DEC-2011/01/D/ST6/07007. The authors would like to thank the reviewers for their valuable comments and suggestions that improved the quality of the paper.

References


