JOURNAL OF MEDICAL INFORMATICS & TECHNOLOGIES Vol. 20/2012, ISSN 1642-6037

survival analysis, survival trees,
decision list, rule induction

Łukasz WRÓBEL[1]

TREE-BASED INDUCTION OF DECISION LIST FROM SURVIVAL DATA

The paper presents an algorithm for induction of decision list from survival data. The algorithm uses a survival tree as the inner learner which is repeatedly executed in order to select the best rule at each iteration. The effectiveness of the algorithm was empirical tested for two implementations of survival trees on 15 benchmark datasets. The results show that proposed algorithm for survival decision list construction is able to induce more compact models than corresponding survival tree without the loss of the accuracy of predictions.

1. INTRODUCTION

Survival analysis concerns the study of the time to the occurrence of an event such as death, disease, machine failure. It is especially widespread in the medical research, but it is also common in engineering (failure time analysis) and social sciences (event-history analysis). One of the main tasks of survival analysis is the modeling of the impact of covariates on survival time. The most common approach to this problem is the Cox proportional hazards regression [8], however data mining techniques have become more and more popular over the last years. Especially useful are the methods which represent a data model in simple to understand and interpret form. The most popular ones are survival trees [4] which are an adaptation of classification and regression trees [5] to the problem of survival. Survival trees can divide the observations into groups with similar survival rate according to the values of covariates.

The other representation of the data model in understandable form is a decision list [22] i.e. an ordered set of if-then rules. The decision list can be more compact and therefore easier to understand than corresponding it tree. This raises the question whether tree-based model can be transformed into (potentially) less complex decision list without the loss of the predictive accuracy of the model. Such works were undertaken, inter alia, in the papers [11, 13] where separate-and-conquer approach was used in order to derive decision list from classification and regression trees. Especially good results of this strategy are therefore the motivation of this paper to apply the aforementioned technique to survival trees. Naturally, the difference between the problem of classification/regression and survival analysis requires a novel approach.

The aim of this paper is to present the algorithm for induction of decision list from survival data. The algorithm uses a survival tree as the inner learner and separate-and-conquer strategy for rule induction. The paper is organized as follows: Section 2 briefly summarizes rule-based methods for survival analysis. Section 3 discusses the problem of survival decision list induction. The experimental verification of the presented algorithm is the subject of Section 4. Section 5 is a summary of the paper and the presentation of the directions of further research.

2. RELATED WORKS

The problem of rule induction from survival data was undertaken in the papers [3, 17, 19]. The paper [19] describes the rough sets hybrid intelligent system for survival analysis. One of the parts of this system is the ELEM2 [1] algorithm which is used in order to generate rules for survival prediction. The ELEM2 is the method for induction of decision rules, therefore, before running the algorithm the decision classes were created by discretization of the survival time. The rough set approach to survival analysis is

---

[1] Silesian University of Technology, ul. Akademicka 16, 44-100 Gliwice, Poland.
email: lukasz.wrobel@polsl.pl.

also presented in [3]. Similarly as in [19], the decision rule induction algorithm was applied, but the ranges of decision classes were determined in a different way – on the basis of the values of a prognostic index (PI) derived from the Cox's proportional hazard model. It should be noted that the approaches presented in [3, 19] induce rules from survival data in indirect way i.e. they apply classical decision rule induction algorithms to the preprocessed (divided into decision classes) survival data. The direct induction of the rules from survival data was presented in the paper [17] in which the authors adapted the Logical Analysis of Data technique [9] for this purpose.

Tree-based methods have noticeably received much more attention in survival analysis than rule-based ones. On the other side, 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. Therefore, survival trees [4] can be also mentioned as related work. The key idea of the application of tree-based techniques to survival data lies in the splitting criterion. The most popular approach is to select a split which maximizes the difference between survival distributions of child nodes, that it is often achieved by the maximization of the log-rank test statistics [7, 18, 23].

3. INDUCTION OF SURVIVAL DECISION LIST

The examples of survival data are usually described by a set $A \cup \{t, \delta\}$ of attributes, where A is a set of conditional attributes (covariates), t is survival time and δ is a censoring status. The survival time attribute contains information about the time that has elapsed since the beginning of observation. The censoring status specifies whether the event occurred or not. In this paper, it is assumed the right-censored model of data which is the most common one. Right-censored examples are those for which event was not observed and their lifetime is greater than recorded time.

Decision list with survival outcome can be represented in the following form (1):

$$\begin{aligned}&\textbf{if } c_{11} \wedge c_{12} \wedge \ldots \wedge c_{1n_1} \textbf{ then } S_1(t)\\ &\textbf{else if } c_{21} \wedge c_{22} \wedge \ldots \wedge c_{2n_2} \textbf{ then } S_2(t)\\ &\ldots\\ &\textbf{else } S_k(t)\end{aligned} \tag{1}$$

Survival decision list is an ordered set of if-then rules i.e. the rules are tested one by one, starting from the top of the list, and the prediction is made on the basis of the first matching rule. The conclusion of the rule consists of the estimate of the survival function. In this paper the survival function is estimated by Kaplan-Meier (KM) method [16] which is one of the fundamental techniques of survival analysis. The premise of the rule is formed by the conjunction of elementary conditions. The elementary condition is the most often the expression of the form *a rel v* where $a \in A$, *rel* is one of the relation symbols from the set $\{=, \leq, \geq, >, <\}$, and v is a numerical or nominal value from the range of the attribute a. The last k-th rule is the default one and it assigns the survival estimate to the examples which are not covered by any of the preceding rules.

The tree-based induction of decision list from survival data follows the separate-and-conquer approach presented in [11, 13]. The outline of the algorithm can be summarized as follows:

```
Input:
  dataset: a training set of examples
  α: a significance level of the log-rank test
Output: decision list
  decisionList := Ø
  uncoveredDataset := dataset
  while (uncoveredDataset ≠ Ø} do
    survivalTree := LearnSurvivalTree(uncoveredDataset)
    rules := LeavesToRules(survivalTree)
    rules := Prune(rules, α)
    bestRule := select from the rules the rule with
                the lowest p-value of the log-rank test
    if (Count(rules) = 1 or LogrankPvalue(bestRule) > α)
```

```
        bestRule := DefaultRule(uncoveredDataset)
      end if
      covered := Covered(bestRule, uncoveredDataset)
      uncoveredDataset := uncoveredDataset \ covered
      decisionList := decisionList ∪ {bestRule}
    end while
    return decisionList
```

In the first step, the survival tree is built from the whole input training set and each path from the root to the leaf of the tree is transformed into the rule in which elementary conditions correspond to tests encountered on this path. Next, each rule obtained in such a way is pruned. The aim of the pruning phase is to maximize the difference between the survival distribution of examples covered by the examined rule and the survival distribution of remaining observations. For this purpose, the elementary conditions are deleted from the rule if it causes the improvement according to the log-rank test [20]. The log-rank test is the most often used one to compare the difference between two survival distributions. The null hypothesis of this test is that there is no overall difference between them. The rule pruning uses a hill climbing strategy. At each iteration, it deletes the elementary condition without which the rule has the lowest p-value of the log-rank test. The removal of conditions is repeated until it does not increase the log-rank p-value of the rule beyond the significance level α specified by a user. Among all pruned rules the rule with the lowest p-value of log-rank test is selected as the final one. If the p-value of the best rule exceeds the significance level α or the survival tree contains only one root leaf then default rule is created with KM estimate based on all remaining observations. The examples covered by the best rule are removed from the training set, the rule is added to the list and the process of rule induction is recursively executed for remaining training observations. The algorithm terminates if all examples of input dataset are covered.

4. EXPERIMENTS

The experiments were carried out independently for two implementations of the survival trees available in the R environment [21]: the RPART algorithm from the R package of the same name [28], and the CTREE algorithm form the package *party* [14]. These algorithms served two functions. Firstly, they were used as the inner learners in the presented decision list induction algorithm, and secondly, they were run as standalone in order to compare results of the survival tree with the corresponding results of decision list. For both implementations, the algorithm for survival decision list construction was run with significance level parameter equal to 0.05. The survival trees were executed in their default configurations.

The survival decision list was compared with the corresponding survival tree according to the size and the integrated Brier score [12]. The size of the tree is calculated as the number of terminal nodes. The size of the decision list is expressed by the number of rules in the list (taking also the default one into account). The integrated Brier score (IBS) is a popular criterion for the evaluation of performance of survival models. The Brier score calculates the squared difference between true event status at given time t^* and the predicted event status $p(t^*)$ at that time. Let t_i and δ_i denote survival time and censoring status respectively ($\delta_i=1$ if the event occurred, and 0 otherwise) of i-th example (i=1...n, where n is the number of examples in the test set). The Brier score for survival data is expressed by the following formula (2):

$$BS(t^*) = \frac{1}{n}\sum_{i=1}^{n}\{[(0-p(t^*)]^2 I(t_i \le t^* \wedge \delta_{\mathrm{i}} = 1)\cdot G^{-1}(t_i) + [(1-p(t^*)]^2 I(t_i > t^*)\cdot G^{-1}(t^*)\} \tag{2}$$

The true event status for i-th example is equal to 0 if the event occurred for this example before or at the time t^*, and it is equal to 1 if the survival time t_i of the example is greater than t^* (the *I* in the formula is the indicator function). The censoring is taken into account by weighting the squared differences by the inverse of the KM estimate *G* of the censoring survival function. The Brier score is a cost-type criterion i.e. the lower the value the more accurate the prediction.

The integrated Brier score summarizes the prediction error over all times in the test set and it is given by (3):

$$IBS = \frac{1}{\max(t_i)} \int_0^{\max(t_i)} BS(t^*)dt^* \tag{3}$$

The results of the comparison are presented in Table 1. The experiments were carried out on 15 survival datasets available in the following R packages: *survival, ipred, penalized, RcmdrPlugin.survival, KMsurv, timereg, quantreg, TwoWaySurvival, epitools, mboost, NestedCohort.* The datasets differ in the number of examples and attributes as well as in the number of censored observations. Presented results are an average from the 10-fold cross validation. Additionally, in the last row of Table 1 there is shown an average value of IBS and size over all 15 datasets for each examined algorithm.

The results of decision list marked with a sign $^{(-)}/^{(+)}$ indicate statistically significant (at 0.05 level) degradation/improvement over the results obtained by the corresponding implementation of the survival tree (unmarked results are not statistically different from the results obtained by tree counterparts). The comparison was made separately for RPART and CTREE. The paired two-sample t-test was used in order to compare algorithms on single data. The statistical comparison over all 15 datasets (the last row of Table 1) was performed with the use of Wilcoxon signed-rank test (as proposed in [10]).

Table 1. Results obtained by the RPART, CTREE survival tree algorithms and their decision list counterparts.

	RPART				**CTREE**			
	tree		**decision list**		**tree**		**decision list**	
dataset	**IBS**	**size**	**IBS**	**size**	**IBS**	**size**	**IBS**	**size**
cancer	0.162	9.4	$0.148^{(+)}$	$3.8^{(+)}$	0.151	3.0	0.150	$2.0^{(+)}$
dlbcl	0.305	2.9	0.296	$2.5^{(+)}$	0.267	1.4	0.275	1.4
gbsg2	0.175	6.2	0.181	6.6	0.177	4.0	0.178	3.2
mgus	0.164	11.3	$0.176^{(-)}$	$2.3^{(+)}$	0.165	5.8	0.170	$3.3^{(+)}$
nki70	0.235	9.6	0.202	$3.4^{(+)}$	0.187	2.2	0.183	2.0
pbc	0.176	13.1	$0.153^{(+)}$	$3.9^{(+)}$	0.163	6.4	$0.143^{(+)}$	$4.0^{(+)}$
rossi	0.115	9.8	$0.073^{(+)}$	$5.7^{(+)}$	0.119	1.9	0.119	1.8
std	0.211	2.5	0.210	$5.0^{(-)}$	0.206	3.6	0.207	$2.1^{(+)}$
trace	0.179	5.7	$0.191^{(-)}$	$4.6^{(+)}$	0.174	13.7	$0.183^{(-)}$	$5.6^{(+)}$
uis	0.144	11.1	0.145	$4.8^{(+)}$	0.147	5.2	$0.142^{(+)}$	$3.1^{(+)}$
unemployed	0.176	5.0	0.173	$6.7^{(-)}$	0.157	3.0	0.159	$2.0^{(+)}$
veteran	0.117	8.4	0.107	$3.1^{(+)}$	0.102	4.0	0.100	$3.0^{(+)}$
wcgs	0.044	5.5	0.043	$8.5^{(-)}$	0.042	9.7	0.042	$4.7^{(+)}$
wpbc	0.240	12.2	$0.182^{(+)}$	$3.6^{(+)}$	0.174	1.9	0.174	1.9
zinc	0.111	10.1	$0.098^{(+)}$	$5.1^{(+)}$	0.105	2.3	0.105	2.1
average	*0.170*	*8.2*	*$0.159^{(+)}$*	*$4.6^{(+)}$*	*0.156*	*4.5*	*0.155*	*$2.8^{(+)}$*

For the RPART implementation, the size of survival decision list was significantly smaller than the size of corresponding survival tree on 11 datasets. Moreover, for this implementation, the decision list significantly improved the IBS on 5 datasets, and the statistically significant degradation was observed only for 2 datasets. In general, for the RPART, the decision list performs much better than its tree counterpart. It is also confirmed by the Wilcoxon test which showed significant improvement in the IBS and size of decision list over the results obtained by survival tree.

The CTREE algorithm generally induces quite compact survival trees, but decision list still allows for further reduction of the size of survival model. The statistically significant decrease in the size was observed for 9 datasets and, similarly as for RPART, the tendency for inducing more compact models by the proposed algorithm was also confirmed by the Wilcoxon test. The statistically significant improvement in the values of IBS was observed for *pbc* and *uis* datasets, however, for the CTREE implementation the Wilcoxon test does not show significant difference in the values of IBS between tree and decision list.

5. CONCLUSIONS AND FUTURE WORK

The paper presented the algorithm for tree-based induction of decision list from survival data. The key idea of the algorithm is to derive rules from a survival tree in separate-and-conquer fashion. The empirical part of this study showed that the representation of survival data in the form of decision list is a good alternative to tree-based models. The decision list is able to represent survival model in more compact form than corresponding survival tree and improve the predictive accuracy of survival model on unseen examples.

The proposed approach executes survival tree learner multiple times in order to generate rules, and consequently it has higher computational complexity than a single survival tree. Therefore, an interesting direction of future works will be to develop an algorithm which will induce decision list without the use of survival tree as the inner learner. It can be achieved by incorporating splitting criteria known from survival trees into the rule learning process. According to the previous works of the author [25-27], the promising path will be also to adapt the rule induction algorithm which uses sequential covering strategy and rule quality measures [2, 6, 15, 24-27] to the problem of rule learning from survival data.

ACKNOWLEDGEMENTS

This work was co-financed by the European Union within the European Social Fund's project "SWIFT" POKL.08.02.01-24-005/10.

BIBLIOGRAPHY

[1] AN, A., CERCONE, N., ELEM2: A learning system for more accurate classifications. In Advances in Artificial Intelligence, MERCER R., NEUFELD E., Eds., Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 1998, Vol. 1418 , pp. 426–441.

[2] AN, A., CERCONE, N., Rule quality measures for rule induction systems: Description and evaluation, Computational Intelligence, 2001, Vol. 17, No. 3, pp. 409–424.

[3] BAZAN J.G., OSMÓLSKI A., SKOWRON A., ŚLĘZAK D., SZCZUKA M.S., WRÓBLEWSKI J., Rough set approach to the survival analysis, Rough Sets and Current Trends in Computing, ALPIGINI J.J., PETERS, J.F., SKOWRONEK J., ZHONG E., Eds., Lecture Notes in Computer Science, Springer, 2002, Vol. 2475, pp. 522–529.

[4] BOU-HAMAD I., LAROCQUE D., BEN-AMEUR H., A review of survival tree,. Statistics Surveys, 2011, Vol. 5, pp. 44–71.

[5] BREIMAN L., FRIEDMAN J. ., OLSHEN R.A., STONE C.J., Classification and Regression Trees, Wadsworth, 1984.

[6] BRUHA I., TKADLEC J., Rule quality for multiple-rule classifier: Empirical expertise and theoretical methodology, Intelligent Data Analysis, 2003, Vol. 7, No. 2, pp. 99–124.

[7] CIAMPI A., THIFFAULT J., NAKACHE J., ASSELAIN B., Stratification by stepwise regression, correspondence analysis and recursive partition: a comparison of three methods of analysis for survival data with covariates, Computational statistics & data analysis, 1986, Vol. 4, No. 3, pp. 185–204.

[8] COX D., Regression models and life-tables, Journal of the Royal Statistical Society, Series B (Methodological) 1972, pp. 187–220.

[9] CRAMA Y., HAMMER P., IBARAKI T., Cause-effect relationships and partially defined boolean functions, Annals of Operations Research, 1988, Vol. 16, No. 1, pp. 299–325.

[10] DEMŠAR J., Statistical comparisons of classifiers over multiple data sets, The Journal of Machine Learning Research, 2006, Vol. 7, pp. 1–30.

[11] FRANK E., WITTEN I., Generating accurate rule sets without global optimization, Proceedings of the Fifteenth International Conference on Machine Learning, Morgan Kaufmann Publishers Inc., 1998, pp. 144–151.

[12] GRAF E., SCHMOOR C., SAUERBREI W., SCHUMACHER M., Assessment and comparison of prognostic classification schemes for survival data, Statistics in medicine, 1999, Vol. 18, pp. 2529–2545.

[13] HOLMES G., HALL M., FRANK E., Generating rule sets from model trees, Twelfth Australian Joint Conference on Artificial Intelligence, Springer, 1999, pp. 1–12.

[14] HOTHORN T., HORNIK K., ZEILEIS A., Unbiased recursive partitioning: A conditional inference framework, Journal of Computational and Graphical Statistics, 2006, Vol. 15, No. 3, pp. 651–674.

[15] JANSSEN F., FÜRNKRANZ J., On the quest for optimal rule learning heuristics, Machine Learning, 2010, Vol. 78, No. 3, pp. 343–379.

[16] KAPLAN E.L., MEIER, P., Nonparametric estimation from incomplete observations, Journal of the American Statistical Association, 1958, Vol. 53, No. 282, pp. 457–481.

[17] KRONEK L.P., REDDY A., Logical analysis of survival data: prognostic survival models by detecting high-degree interactions in right-censored data, Bioinformatics 2008, Vol. 24, No. 16, pp. 248–253.

[18] LEBLANC M., CROWLEY J., Survival trees by goodness of split, Journal of the American Statistical Association, 1993, pp. 457–467.

[19] PATTARAINTAKORN P., CERCONE N., A foundation of rough sets theoretical and computational hybrid intelligent system for survival analysis, Computers & Mathematics with Applications, 2008, Vol. 56, No. 7, pp. 1699–1708.

[20] PETO R., PETO J., Asymptotically efficient rank invariant test procedures, Journal of the Royal Statistical Society, Series A (General), 1972, pp. 185–207.

[21] R DEVELOPMENT CORE TEAM, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, 2011. ISBN 3-900051-07-0.

[22] RIVEST R.L., Learning decision lists, Machine Learning, 1987, Vol. 2, pp. 229–246. 10.1007/BF00058680.

[23] SEGAL M., Regression trees for censored data, Biometrics, 1988, pp. 35–47.

[24] SIKORA M., Rule quality measures in creation and reduction of data rule models, Rough Sets and Current Trends in Computing, GRECO S., HATA Y., HIRANO S., INUIGUCHI M., MIYAMOTO S., NGUYEN H., SŁOWINSKI R., Eds., Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2006, Vol. 4259, pp. 716–725.

[25] SIKORA M., SKOWRON A., WRÓBEL Ł., Rule quality measure-based induction of unordered sets of regression rules, Artificial Intelligence: Methodology, Systems and Applications, RAMSAY A.,. AGRE G., Eds., Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2012, Vol. 7557, pp. 162–171.

[26] SIKORA M., WRÓBEL Ł., Data-driven adaptive selection of rules quality measures for improving the rules induction algorithm, Rough Sets, Fuzzy Sets, Data Mining and Granular Computing, KUZNETSOV S., SLEZAK D., HEPTING D., MIRKIN B., Eds., Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2011, Vol. 6743, pp. 278–285.

[27] SIKORA M., WRÓBEL Ł., Data-driven adaptive selection of rule quality measures for improving rule induction and filtration algorithms, International Journal of General Systems (in press).

[28] THERNEAU T., ATKINSON B., RIPLEY B., RPART: Recursive partitioning, R package version 3.1-54, R port by Brian Ripley, 2012.