An Intelligent Medical Treatment Recommendation System Based on a Random Forest Cascade in the Framework of Survival Analysis

Lev V. Utkin[1], Mikhail A. Ryabinin[2]
Department of Telematics
Peter the Great St.Petersburg Polytechnic University
St. Petersburg, Russia
[1]lev.utkin@gmail.com, [2]mihail-ryabinin@yandex.ru

Anna A. Meldo
Department of Radiology
St. Petersburg Clinical Scientific-Practical Center of Specialized Types of Medical Care (Oncology)
St. Petersburg, Russia
anna.meldo@yandex.ru

***Abstract*— The paper proposes a new architecture of the recommendation intelligent system, which allows choosing a personal optimal treatment for a patient on the basis of his medical records in terms of minimizing the hazard function. The proposed approach for implementing the system is based on the statistical survivor analysis. The calculation of the risk function is carried out using a cascade of random survival forests such that every random forest implements the expansion of the well-known Cox proportional hazard model. Random survival forests are trained using patient data and differ from conventional regression random forests by a specific splitting function that maximizes the difference between hazard functions of two split sets of training data. A peculiarity of the proposed architecture is the random forest cascade organization which can be regarded as one of the modifications of the deep forest.**

Keywords*— *survival analysis; random forest; artificial intelligence; hazard function; patient; Cox model

I. INTRODUCTION

A lot of computer aided diagnosis (CAD) systems have been developed in order to provide successful detection of a disease and to facilitate making decision to start treatment process at early stage. Most CAD systems aim to detect a disease or its features. However, there are a few systems which take into account survival aspects of a patient especially of a cancer patient. A large amount of data has been recorded about patients, their peculiarities in hospitals. It is important to utilize the data and to develop CAD systems which could help to a doctor to choose a proper treatment for every patient on the basis of his or her state and disease factors. The correct and early diagnosing of a cancer may save the patient life. So the information about patients and their disease is essential to help.

A basis for such the CAD systems may be survival analysis or time-to-event analysis which can be regarded as a fundamental tool which is used in many applied areas. One of the most important areas is the medical research where survival models are widely used to evaluate the significance of prognostic variables in outcomes such as death or cancer recurrence and subsequently inform patients of their treatment options [14]. The data set used in the survival analysis or just the survival data differ from many data sets by the fact that time to event of interest for a part of observations or patients is unknown because the event might not have happened during the period of study. If the observed survival time is less than or equal to the true survival time, then we have a special case of censoring data called right-censoring data.

The survival models can be divided into three parts: parametric, nonparametric and semiparametric. It is assumed in parametric models that the type of the probability distribution of survival times is known, for example, the exponential, Weibull, gamma distributions. One of the simplest survival models is the Kaplan-Meier estimator which is a non-parametric model used to compute the survival function of a homogeneous data set, i.e., the model does not take into account the fact that the patients may differ by their features.

A popular regression model for the analysis of survival data is the well-known Cox proportional hazards model, which is a semi-parametric model that calculates the effects of observed covariates on the risk of an event occurring, for example, the death or failure [4]. The proportional hazards assumption in the Cox model means that different patients have hazard functions that are proportional, i.e., the ratio of the hazard functions for two patients with different prognostic factors or covariates is a constant and does not vary with time. The Cox model is a very powerful method for dealing with survival data. As a result, a lot of approaches dealing with the Cox model and its modifications have been proposed last decades. A clear taxonomy of survival analysis methods and their comprehensive review is presented by Wang et al. [21].

It should be noted that the Cox model may provide unsatisfactory results under conditions of a high dimensionality of survivor data and a small number of observations, for example, when we deal with gene expression data. To overcome this problem, Tibshirani [18] proposed a modification of the Cox model based on the Lasso method. Another problem of the Cox model is the linear relationship assumption between covariates and the time of event

This work is supported by the Russian Science Foundation under grant 18-11-00078.

978-1-5386-5824-6/18/$31.00 ©2018 IEEE

occurrence. Various modifications have been proposed taking into account the corresponding non-linear relationship between covariates and the time of event. Faraggi and Simon in their pioneering work [6] presented an approach to modelling survival data using the input-output relationship associated with a simple neural network as a basis for a non-linear proportional hazards model. The model was a basis for developing a generalization using the deep neural networks [14], [17]. Several models based on neural networks are considered in the review [21]. The use of neural networks requires a lot of survival data. Therefore, Van Belle et al. [20] proposed to use SVM in order to enhance the model by the small amount of training data.

Another approach for dealing with the limited survival data is to use survival trees and the survival random forests. Due to many advantages of decision trees as a tool for classification and regression, several tree-based modifications solving the survival analysis problems have been proposed last decades [3],[7],[15]. A detailed review of survival trees as well as random survival forests is represented by Bou-Hamad et al. [1].

Random forests were introduced by Breiman [2] in order to overcome some shortcomings of the decision trees. It turns out that the random forests became a very powerful, efficient and popular tool for the survival analysis. The popularity of random survival forests stems from many useful factors. First of all, Ishwaran and Kogalur [12] pointed out that the random forests require only three tuning parameters to be set (a number of randomly selected predictors, a number of trees grown in the forest, and a splitting rule).

As a result, a lot of models based on random forest have been developed for dealing with survival data [1], [10], [11]. Most models are very similar and differ in splitting criteria and the ensemble estimation. Most survival random forests use averaging of the tree cumulative hazard estimates.

Since the random survival forest is one of the most efficient models in survival analysis, we pay attention to this model and propose an approach for its improving. A main idea underlying the approach is to construct a random survival forest cascade which can be viewed as a special case of the deep forest proposed by Zhou and Feng [23]. In fact, we develop a deep survival forest. The crucial point in the proposed cascade structure is the stacking algorithm implementation. The stacking method was presented by Wolpert [22] to realize an idea that the later layers of a multilayer structure can learn the mistakes that classifiers in the previous layers make, and correct them. Therefore, we also propose a modification of the stacking algorithm taking into account peculiarities of the random survival forests.

II. The Cox model

In survival analysis, a patient *i* is represented by a triplet $(\mathbf{x}_i, \delta_i, T_i)$, where $\mathbf{x}_i = (x_{i1},...,x_{im})$ is the vector of the patient characteristics (features); T_i indicates time to event of the patient. If the event of interest is observed, T_i corresponds to the time between baseline time and the time of event happening, in this case the event indicator is $\delta_i = 1$. We have an uncensored observation. If the event is not observed, then T_i corresponds to the time between baseline time and the end of the observation, and $\delta_i = 0$. We have a censored observation. Suppose a training set *D* consists of *n* triplets $(\mathbf{x}_i, \delta_i, T_i)$, $i = 1,...,n$. Survival analysis aims to estimate the time to the event of interest *T* for a new patient with feature vector $\mathbf{x}$ by using the set *D*.

The survival and hazard functions are key concepts in survival analysis. The survival function denoted by $S(t)$ as a function of time *t* is the probability of surviving up to that time, i.e., $S(t) = \Pr\{T > t\}$. The hazard function $h(t)$ is the rate of event at time *t* given that no event occurred before time *t*. The survival function is determined through the hazard function as

$$S(t) = \exp\left(-\int_0^t h(x)\mathbf{d}x\right).$$

According to the Cox proportional hazards model [4], [9], the hazard function given predictor values $\mathbf{x}$ is defined as

$$h(t \mid \mathbf{x}) = h_0(t) \cdot \Psi(\mathbf{x}, \mathbf{b}) = h_0(t) \cdot \exp\left(\psi(\mathbf{x}, \mathbf{b})\right).$$

Here $h_0(t)$ is a baseline hazard function; $\Psi(\mathbf{x})$ is the risk function; $\mathbf{b} = (b_1,...,b_m)$ is an unknown vector of regression parameters. The reparametrization $\Psi(\mathbf{x}, \mathbf{b}) = \exp\left(\psi(\mathbf{x}, \mathbf{b})\right)$ is used in the Cox model. The function $\psi(\mathbf{x}, \mathbf{b})$ in the model is linear, i.e., $\psi(\mathbf{x}, \mathbf{b}) = \mathbf{x}\mathbf{b}^{\mathbf{T}}$.

The idea underlying the use of neural networks in survival analyzing is to replace the linear function $\psi(\mathbf{x})$ with a non-linear function realized by means of a neural network [6].

Since patients are subject to different levels of risk based on their relevant prognostic features and their treatment, it is interesting to find out the best treatment. We also consider an approach proposed by Katzman et al. [14]. Suppose that all patients belong to one of *s* treatment groups $\tau \in \{1,...,s\}$. It is assumed that each treatment *i* has an independent risk function $\exp\left(\psi_i(\mathbf{x})\right)$ and the baseline hazard function $h_0(t)$ is the same for all treatment. One of the ways to find the best treatment is to compute the recommender function which is defined as

$$rec_{ij}(\mathbf{x}) = \log\left(\frac{h(t;\mathbf{x} \mid \tau = i)}{h(t;\mathbf{x} \mid \tau = j)}\right) = \psi_i(\mathbf{x}) - \psi_j(\mathbf{x}).$$

To provide personalized treatment recommendations in accordance with the above function, we compute functions $\psi_i(\mathbf{x})$ and $\psi_j(\mathbf{x})$ corresponding to different treatment groups. If the obtained function $rec_{ij}(\mathbf{x})$ is positive, then treatment *j* is preferable in comparison with treatment *i*. In the case of a negative recommender function, treatment *i* is more effective and leads to a lower risk than treatment *j*.

To compare the survival models, the C-index proposed by Harrell et al. [8] is used. It estimates how good the model is at ranking survival times. In fact, this is the probability that the event times of a pair of patients are correctly ranking.

We consider admissible pairs $\{(\mathbf{x}_i, \delta_i, T_i), (\mathbf{x}_j, \delta_j, T_j)\}$ for $i \le j$ in *D*. Then the C-index is calculated as the ratio of the number of pairs correctly ordered by the model to the total number of admissible pairs *M*. A pair is not admissible if the earliest time in the pair is censored. If the C-index is equal to 1, then the corresponding survival model is supposed to be perfect. If the C-index is 0.5, then the model is no better than random guessing. Let $t_1^*, ..., t_q^*$ denote predefined time points, for example, $t_1, ..., t_N$, where *N* is the number of events. If the output of a survival algorithm is the predicted survival function $S(t)$, then the C-index is formally calculated as [21]:

$$C = \frac{1}{M} \sum_{i:\delta_i=1} \sum_{j:t_i<t_j} \mathbf{1}\left[S(t_i^* \mid \mathbf{x}_i) > S(t_j^* \mid \mathbf{x}_j) \right].$$

Here $\mathbf{1}[a]$ is the indicator function taking the value 1 if *a* is true, and 0 otherwise; *S* is the estimated survival function.

There are different definitions of the C-index, which depend on the output of a survival algorithm. However, we will use the above definition which plays an important role in the proposed scheme of the stacking algorithm.

III. Random Survival Forests

It has been mentioned that the random survival forest is one of the best models for survival analysis due to its properties. This is the main reason for its modifying to improve the survival analysis results and to increase the prediction accuracy.

A general algorithm for constructing random survival forests can be represented as follows [13]:

1. Draw *Q* bootstrap samples from the original data. Note that each bootstrap sample excludes on average 37% of the data, called out-of-bag data (OOB data).
2. Grow a survival tree for each bootstrap sample. At each node of the tree, randomly select $m^{1/2}$ candidate variables. The node is split using the candidate variable that maximizes survival difference between daughter nodes.
3. Grow the tree to full size under the constraint that a terminal node should have no less than $d > 0$ unique deaths.
4. Calculate a cumulative hazard function for each tree. Average to obtain the ensemble cumulative hazard function.
5. Using OOB data, calculate prediction error for the ensemble cumulative hazard function.

An important question of random survival forests, which defines their different implementations is the splitting rule. As shown by Ishwaran et al. [13], a good split maximizes survival difference across the two sets of data. There are many splitting rules, but we point out three rules: (1) the rule based on the log-rank test for a split; (2) the conservation of events splitting; (3) the approximate log-rank splitting. Every rule has pros and cons. A detailed review of the rules can be found in [13], [21].

Let $\{t_{j,k}\}$ be the distinct death times in node *k* of the *q*-th tree, and $Z_{j,k}$ and $Y_{j,k}$ equal the numbers of deaths and patients at risk at time $t_{j,k}$. The cumulative hazard estimate for node *k* is defined as

$$H_k(t) = \sum_{t_{j,k} \le t} Z_{j,k} / Y_{j,k}.$$

If the *i*-th patient with features $\mathbf{x}_i$ falls into node *k*, then we can say that $H(t \mid \mathbf{x}_i) = H_k(t)$. The ensemble cumulative hazard estimate for the *i*-th patient is obtained by averaging cumulative hazard estimates of all Q trees, i.e.,

$$H_{\text{forest}}(t \mid \mathbf{x}_i) = \frac{1}{Q} \sum_{q=1}^{Q} H_q(t \mid \mathbf{x}_i).$$

The survival function can be obtained from the cumulative hazard estimates. Another ensemble estimate is considered by Ishwaran et al. [13], where OOB data are used.

IV. A Random Survival Forest Cascade

We consider a random forest cascade as a special case of the deep forest proposed by Zhou and Feng [23]. The deep forest is a cascade forest structure where each level of a cascade receives feature information processed by its preceding level, and outputs its processing result to the next level [23]. We suppose that the number of levels in the cascade is K

One of the important ideas underlying the cascade forest structure is the concatenation of the output of the forest from every level of the cascade with the original vector to be input to the next level. This idea can be viewed as a type of the stacking method [22]. The stacking algorithm trains the first-level learners using the original training data set and then it generates a new data set for training the second-level learner (meta-learner) from the outputs of the first-level learners. In contrast to the stacking algorithm, the deep forest simultaneously uses the original vector and the output data at the next cascade level by means of their concatenation. In other words, the feature vector is enlarged after every cascade level.

The main difficulty for implementing the stacking algorithm is to choose some representation of the random survival forest output. This representation should be compact and informative simultaneously.

The output of the random survival forest is the hazard function or the survival function. We cannot concatenate it with the original data $\mathbf{x}_i$ as augmented features in order to implement the stacking scheme because the augmented features will mask the original data in this case. Therefore, we propose to concatenate two features. The first feature is the mean time to the event m_i of the *i*-th patient, which can be simply computed from the survival function by means of its integrating. The second feature is more complex. We call it a partial C-index for the *i*-th patient, and denote it as C_i. The

partial index is defined in the following way. If $\delta_i = 0$, then $C_i = 0$. If $\delta_i = 1$, then there holds

$$C_i = \frac{1}{M_i} \sum_{j:t_i<t_j} \mathbf{1}\left[S(t_i^* \mid \mathbf{x}_i) > S(t_j^* \mid \mathbf{x}_j) \right].$$

Here M_i is the number of admissible pairs for the *i*-th patient. One can see from the above expression that the partial C-index shows how the prediction concerning the *i*-th patient is concordant with all patients which are form the admissible pairs with the *i*-th patient. In fact, this is a measure of the prediction quality of a single patient. Finally, the input of the next level of the forest cascade can be represented as follows:

$$\mathbf{x}_i^{k+1} \leftarrow (\mathbf{x}_i^k, m_i^k, C_i^k).$$

The upper index *k* means the cascade level, $k = 1,...,K$.

For a new patient with features $\mathbf{x}$, we obtain the hazard function at the last level of the trained cascade. By using the expression for the recommender function $rec_{ij}(\mathbf{x})$, we get the personalized treatment recommendations for the new patient. Now we use non-simplified expression for $rec_{ij}(\mathbf{x})$, i.e.,

$$rec_{ij}(\mathbf{x}) = \log\left(\frac{h(t;\mathbf{x} \mid \tau = i)}{h(t;\mathbf{x} \mid \tau = j)} \right).$$

The hazard function $h(t;\mathbf{x} \mid \tau = i)$ is the derivative of the cumulative hazard function $H_{\text{forest}}(t \mid \mathbf{x}^K)$ obtained for the treatment group $\tau = i$.

V. Conclusion

The recommendation intelligent system, which allows choosing a personal optimal treatment for a patient based on the application of a cascade of the random survival forests has been proposed. The main contribution is that a new algorithm of stacking in the forest cascade is proposed. It consists in concatenation of original feature vector of a patient and two features obtained from the output of the level forest. The proposed implementation of the system can be regarded as a first attempt to construct the deep survival forest. In order to improve the proposed structure, we can introduce a procedure which assigns weights to trees in every forest in an optimal way similar to the same procedure used in the Siamese deep forest [19]. However, this is a direction for further research.

References

[1] Bou-Hamad I., Larocque D., Ben-Ameur H.. A review of survival trees. Statistics Surveys. 5:44-71. 2011.

[2] Breiman L.. Random forests. Machine learning. 45(1). 5-32. 2001.

[3] Ciampi A. Generalized regression trees. Computational Statistics & Data Analysis. 12:57-78. 1991.

[4] Cox D.R. Regression models and life-tables. Journal of the Royal Statistical Society, Series B (Methodological). 34(2).187-220. 1972.

[5] Devarajn K., Ebrahimi N. A semi-parametric generalization of the cox proportional hazards regression model. Inference and applications. Computational Statistics & Data Analysis. 55(1). 667-676. 2011.

[6] Faraggi D., Simon R. A neural network model for survival data. Statistics in medicine. 14(1). 73-82. 1995.

[7] Gordon L., Olshen R.A. Tree-structured survival analysis. Cancer treatment reports. 69(10). 1065-1069. 1985.

[8] Harrell F., Califf R., Pryor D., Lee K., Rosati R. Evaluating the yield of medical tests. Journal of the American Medical Association. 247. 2543-2546. 1982.

[9] Hosmer D., Lemeshow S., May S.. Applied Survival Analysis. Regression Modeling of Time to Event Data. John Wiley & Sons. New Jersey. 2008.

[10] Hu C., Steingrimsson J.A. Personalized risk prediction in clinical oncology research: Applications and practical issues using survival trees and random forests. Journal of Biopharmaceutical Statistics. 28(2). 333-349. 2018.

[11] Ishwaran H., Blackstone E.H., Pothier C.E., Lauer M.S.. Relative risk forests for exercise heart rate recovery as a predictor of mortality. Journal of the American Statistical Association. 99. 591-600. 2004.

[12] Ishwaran H., Kogalur U.B. Random survival forests for R. R News. 7(2). 25-31. 2007.

[13] Ishwaran H., Kogalur U.B., Blackstone E.H., Lauer M.S.. Random survival forests. Annals of Applied Statistics, 2:841-860, 2008.

[14] Katzman J.L., Shaham U., Cloninger A., Bates J., Jiang T., Kluger Y. Deepsurv: Personalized treatment recommender system using a Cox proportional hazards deep neural network. BMC medical research methodology, 18(24):1-12, 2018.

[15] LeBlanc M., Crowley J. Relative risk trees for censored survival data. Biometrics, 48(2):411-425, 1992.

[16] Lee E.T., Wang J.W. Statistical Methods for Survival Data Analysis. John Wiley & Sons. New Jersey. 2003.

[17] Nezhad M.Z., Sadati N., Yang K., Zhu D. A deep active survival analysis approach for precision treatment recommendations: Application of prostate cancer. arXiv. 1804.03280, April 2018.

[18] Tibshirani R. The lasso method for variable selection in the cox model. Statistics in medicine, 16(4):385-395, 1997.

[19] Utkin L.V., Ryabinin M.A. A Siamese deep forest. Knowledge-Based Systems, 139:13-22, 2018.

[20] Van Belle V., Pelckmans K., Suykens J.A.K., Van Huffel S. Support vector machines for survival analysis. Proceedings of the Third International Conference on Computational Intelligence in Medicine and Healthcare (CIMED2007). Pp. 1-8. 2007.

[21] Wang P., Li Y., Reddy C.K. Machine learning for survival analysis: A survey. arXiv:1708.04649, August 2017.

[22] Wolpert D.H. Stacked generalization. Neural networks, 5(2):241-259, 1992.

[23] Zhou Z.-H., Feng J. Deep forest: Towards an alternative to deep neural networks. arXiv:1702.08835v2, May 2017.