Multi-Event Survival Prediction for Amyotrophic Lateral Sclerosis

Notebook for the iDPP Lab on Intelligent Disease Progression Prediction at CLEF 2022

Corrado Pancotti[1], Giovanni Birolo[1], Tiziana Sanavia[1], Cesare Rollo[1] and Piero Fariselli[1]

[1]*Dept. of Medical Sciences, University of Turin*

Abstract

ALS is a neurodegenerative disease that causes progressive loss of motor skills, and leads to difficulties in breathing, speaking, swallowing and eventually death, usually in a few years. Despite the lack of treatments, interventions such as non-invasive mechanical ventilation and percutaneous endoscopic gastrostomy can be made to prolong life expectancy when needed. Hence it would be clinically relevant to predict the patients' need of such interventions. To this aim, the Intelligent Disease Progression Prediction challenge was organized, in which partecipants were tasked with developing new methods for risk and time-to-event prediction based on demographical and clinical features. Specifically, the challenge tasks consisted of predicting multiple competing risks, all related to ALS disease progression. We employ several machine learning methods generally applied to survival analysis and classification tasks, some of which are specialized for handling competing risks. All models were optimized through a cross-validation procedure and finally evaluated on an internal test set. The three best performing methods, namely Deep Survival Machines, Gradient boosted regression trees and Time-Aware Classifier Ensemble were selected and submitted to the IDPP challenge at CLEF 2022. The results of the competition showed that our methods achieve on average a c-index of ~ 0.70 and ~ 0.74, using data at time zero and up to six months, respectively.

Keywords

Survival Prediction, Machine Learning, Amyotrophic Lateral Sclerosis

1. Introduction

Here we describe our work for the Intelligent Disease Progression Prediction challenge which is part of CLEF 2022. The goal of the challenge is the prediction of adverse events in amyotrophic lateral sclerosis (ALS) patients. ALS is a neurodegenerative disease that cause the loss of motor neurons, leading to progressive difficulties in movements, speaking, breathing and swallowing and finally to death, often in just two to four years. While no cure exists, common treatments include non-invasive ventilation (NIV) and percutaneous endoscopic gastrostomy (PEG), that are employed to increase life expectancy in patients with severe respiratory or feeding issues.

CLEF 2022: Conference and Labs of the Evaluation Forum, September 5–8, 2022, Bologna, Italy

corrado.pancotti@unito.it (C. Pancotti); giovanni.birolo@unito.it (G. Birolo); tiziana.sanavia@unito.it (T. Sanavia); cesare.rollo@unito.it (C. Rollo); piero.fariselli@unito.it (P. Fariselli)

0000-0003-2327-1148 (C. Pancotti); 0000-0003-0160-9312 (G. Birolo); 0000-0003-3288-0631 (T. Sanavia); 0000-0001-6093-1454 (C. Rollo); 0000-0003-1811-4762 (P. Fariselli)

The need for either NIV or PEG, together with death, are the adverse events to be predicted in the challenge. Specifically, we addressed the first two tasks in the challenge, requiring to rank patients by their risk of impairment (Task 1) and to predict the time of impairment (Task 2), with three possible combinations of impairments:

- a: NIV or death, whichever occurs first;
- b: PEG or death, whichever occurs first;
- c: death.

See the overview papers for a complete overview of the challenge [1, 2].

While standard survival analysis methods are well suited to risk and time-to-event prediction, the challenge outcomes include multiple event types that are all linked to the disease progression. For this reason, the risks of the different event types are not statistically independent, that is, they are competing risks. A common strategy in the presence of multiple events is to analyze each event separately, treating the other events as censoring. However, in the competing risk scenario, this approach is not statistically sound, as the assumption of independent censoring does not hold. For the challenge we choose some methods that explicitly handle competing risks and some that do not, since we are mainly interested in prediction performance and even a statistically unsound method can be highly predictive on a given dataset.

We considered three main approaches. The simplest one consisted on fitting a standard survival predictor separately for each event as outlined above for independent events. Another was the recently developed Deep Survival Machine [3], based on deep learning and capable of handling competing risks. Finally, we also introduced a (possibly novel) time-aware classifier ensemble method, that also handles competing risks.

The paper is organized as follows: Section 2 describes the predictive methods we used in the challenge; Section 3 explains our data preprocessing and model optimization and evaluation strategy; Section 4 reports the performance of the best models that were selected for the challenge; finally, Section 5 summarizes the results.

2. Methodology

Three models were used: deep survival machines, naive multiple event survivals and time aware classifier ensemble. Each model can handle multiple events: $e_1, \ldots, e_n$. The challenge required to predict either a risk rank or a time-to-event and, in both cases, the most likely event.

2.1. Naive Multiple Event Survival (NMES)

A mini-ensemble of survival models, with one model fitted for each event. For each event a model was fitted by considering other events as censored. Note that this approach is statistically sound only if the events are independent, which is unlikely in this case. Then, a final prediction was produced by choosing the model predicting the shortest time-to-event and the corresponding event.

The base method for the ensemble were taken from those available in the scikit-survival python package (v0.17.0)[4], since they either predict a time-to-event or a survival function.

Some methods did not yield a time-to-event directly, but a survival function. In this case the time-to-event was chosen as the time when the survival was equal to 0.5.

The base models we tested were a Cox Regression with ElasticNet penalty, a Gradient-boosting with regression trees[5, 6, 7] and Random Survival Forest[8] (`CoxnetSurvivalAnalysis`, `GradientBoostingSurvivalAnalysis`, `RandomSurvivalForest` in scikit-survival, respectively).

The predicted time-to-event $t > 0$ was converted into an event risk $0 < r < 1$ using the sigmoid function:

$$r = \frac{1}{1 + e^t}$$

2.2. Deep Survival Machines (DSM)

Deep Survival Machines (DSM) [3] are a fully parametric deep learning regression model that can estimate relative risks in a time-to-event prediction problem with censored data, managing also competing events. DSM estimates the conditional survival function $S(t|X)$ (where X represent the covariates) as a mixture of K individual parametric survival distributions (Log-normal or Weibull).

While DSM can handle competing risks, it does not predict which event is the most likely to happen first. In training all events are used, but only one is considered the "main" event and the model gives a prediction on that event. Thus, in order to give an event prediction as required by the challenge for the datasets with multiple events, we followed the same approach as in NMES: we trained two DSM models, one for each event. Finally the two models were put together to obtain a final prediction. For the Task 1, the ensemble prediction resulted in assigning to each patient the event label and the associated risk, based on the highest risk score obtained in the two independent models. For the Task 2 the ensemble prediction resulted in assigning the event label and time-to-event according to which of the two survival functions first reached time equal to 0.5.

DSM model parameters considered during the optimization process were the type of underlying distribution, the number of mixture distributions, the number of layers and their neurons, the learning rate and the batch size.

The hyparameters were choosen in a validation set based on the C-index metric (see section 3.2).

2.3. Time-Aware Classifier Ensemble (TACE)

This methods uses an ensemble of multiclass classifiers to predict an event risk score and the most probable event.

A time t was randomly chosen and a model m_t was fitted on the following subdataset: individuals that were censored before time t were dropped, individuals for which an event occurred before t were labeled with that event and individuals that had and event or were censored after t were labeled as "no event". We assume that the classifier m_t yields a probability vector y_t of length $n + 1$, where the last entry $y_t, n + 1$ is the probability of a "no event".

We averaged the probability of each event across the classifiers to obtain a final score vector $\tilde{y} = \sum_t y_t/N$ with an entry for each event and for the "no event" classes. The final risk is the

sum of event classes $\sum_{i=1}^{n} \tilde{y}_i = 1 - \tilde{y}_{n+1}$ and the predicted event is the most likely one e_i where $i = \arg\max_{j=1,\ldots,n} \tilde{y}_j$.

This method does not yield a time-to-event and thus was only used in Task 1.

3. Experimental Setup

3.1. Data cleaning preprocessing

Each dataset had a training set comprising a table of static features, a table of longitudinal (temporal) features and a table of outcomes. The test set was with same without the table of outcomes.

Static features with more than 90% of missing values in the training set were dropped. Categorical features and boolean features with missing values were one-hot-encoded. Of the resulting encoded features, those with more than 99% of either zeros or ones in the training set were dropped.

While static features are collected once for each individual, longitudinal features are collected repeatedly at different time points for the same individual. A variable had thus a different number of values (possibly also one or zero) for each individual. Since we did not use predictive methods that could handle longitudinal data directly, we re-coded the multiple values that were available for each individual for a longitudinal feature as six static features: the number and the standard deviation of the values and the maximum, minimum, first (lowest collection time) and the last (highest collection time) value.

3.2. Model evaluation and hyperparameter selection

Outcomes were the risk of event and the time window (in six month intervals) for Task 1 and Task 2, respectively. Moreover for datasets A and B the outcomes included also the first event between NIV and death and PEG and death, respectively. For evaluating risk (a regression task) we used mainly the concordance index (c-index) metric. For the NMES models, hyper-parameter selection was performed twice with different metrics: one model was selected only by its c-index and a second model by a "combo" metrics, a weighted linear combination of c-index (60%), accuracy (30%) and a custom interval metric (10%). For the multi-event datasets, accuracy was computed on the prediction of the first event (ignoring censored individuals). For the time windows prediction (a multiclass task) we implemented a custom interval metric that assigns a lower weight to misclassifications where the predicted time-intervals is close to the correct one.

In order to select the best hyper-parameters, we used the following cross-validation strategy. The challenge training set was split into inner training and test set (80-20%). Hyper parameter optimization was performed in cross-validation on the inner training by a grid search. The parameter grids for the different methods are reported in Table 1. The risk prediction performance of each model was then estimated on the inner test set by the c-index metrics and the best performing models were refitted on the whole challenge training, run on the challenge test and submitted for Task 1. The predicted time-to-event from the same models selected for Task 1 were also used for Task 2 (with the exception of TACE).

Table 1
Hyperparameters for used in model optimization. Note that the first three methods (ElasticNet Cox, Gradient Boosting and Random Survival Forest), were used as base predictors in the NMES method. Random Forest was instead the base classifiers we used in the TACE models.

Method	Hyperparameter	Grid Search Values
ElasticNet Cox	`l1_ratio`	$0.01, 0.1, 0.2, 0.5, 0.8, 0.9, 1.0$
Gradient Boosting	`loss`	`coxph, squared, ipcwls`
	`learning_rate`	$0.1, 0.01$
	`n_estimators`	200
	`max_depth`	$1, 2, 3, 4$
	`max_features`	`sqrt, None`
Random Survival Forest	`max_depth`	$2, 3, 4$
	`n_estimators`	200
	`max_features`	`sqrt, None`
Random Forest	`max_depth`	$2, 3, 4$
	`n_estimators`	$200, 400, 600, 800, 1000$
	`max_features`	$10, 20, 30, 40, 50$
	`class_weight`	`balanced, None`
Deep Survival Machines	`distribution_type`	$Weibull, LogNormal$
	`n_mix_distribution`	$1, 2, 3, 4, 5$
	`n_layers`	$1, 2, 3, 4$
	`n_nodes`	$50, 100, 150, 200$
	`l_rate`	$0.0001, 0.001, 0.01$
	`batch_size`	$32, 64, 128$

This strategy was followed for all subtasks and methods except for DSM and the TACE, where because of time constraints we selected optimal hyper parameters only for subtask A0 and reused them in the other subtasks.

4. Results

The challenge objectives were the prediction of adverse event risk (Task 1) and the time of event occurrence (Task 2). Each task comprised three datasets, each one with a different selection of adverse events:

A non-invasive ventilation (NIV) or death,
B percutaneous endoscopic gastrostomy (PEG) or death,
C death.

For A and B only the first occurring event was reported for each patient. For each dataset, two subset of features were considered: one with all the available features and one where only the first ALSFRS-R questionnaire was retained, together with any spirometry data collected before the questionnaire administration. The first subset was denoted by 0 (features up to time zero) and the second by 6 (features for the first six months).

The performance of the submitted models for each task and dataset are reported in Table 2. In particular, for task 1 the c-index is reported both for the internal and the challenge test set (released at the end of the competition); for task 2, mean specificity and recall calculated on the challenge test set are shown.

The submitted models are build as described in section 2 and were optimized as described in section 3. DSM are the Deep Survival Machine models, NMES-CI and NMES-CS are the Naive Multiple Event Survival models, selected by their c-index and combo score on the internal test, respectively, and TACE-RF is the Time Aware Classifier Ensemble using Random Forests as the base classifier. Note that since TACE does not yield a time-to-event, we could not use it in task 2.

In the NMES models, that were optimized separately for each dataset, the best underlying method proved to be the Gradient-Boosting with regression trees, outperforming penalized Cox and Random Survival Forest.

For the task 1, all methods seemed to benefit from the additional six months of longitudinal data: the performance is higher (c-index ~ 0.74) with data up to six months than with data only at time zero (c-index ~ 0.70). On average, there was an increase of about $0.3 - 0.4$ in c-index using data up to six months compared to data at time zero. Also for the task 2 the use of data up to six months seemed to be beneficial. However all models were affected by a strong imbalance: they were very specific (~ 0.86 on average) but not very sensitive, with a low average recall score.

All models achieved comparable performances; there was no model that resulted significantly better than the others. Only the TACE models appeared to be slightly worse than the rest in A6 and B6 datasets.

5. Conclusions and Future Work

In the present work, we have shown the workflow adopted for the Intelligent Disease Progression Prediction challenge. Different models were implemented in a survival analysis scenario for the prediction of patients' risk related to interventions such as non-invasive ventilation (NIV) or percutaneous endoscopic gastrostomy (PEG) and death as well as their time of occurrence. To handle a multiple event prediction scenario, we proposed several machine learning models generally applied to survival analysis and classification tasks, some of which are capable of handling competing risks. For each event a different model was fitted by considering other events as competing (DSM) or censored (NMES); then, a final prediction was produced by choosing the model predicting the shortest time-to-event or the highest risk based on the task considered and the corresponding event. The three best performing methods, on the internal test set, namely Deep Survival Machines, Naive Multiple Event Survivals and Time-Aware Classifier Ensemble were selected and submitted to the IDPP challenge at CLEF 2022.

A clear trend was that models performed better on the six month dataset, for all methods. On the other hand, when looking at each dataset, the models performance was fairly similar, with the exception of the TACE models that lagged behind slightly in A6 and B6 datasets.

Even though we believe to be in an actual competing risk scenario, we observed no clear advantage of the DSM models, that specifically handles competing risks, with respect to the

Table 2
The performance of the submitted models on the internal test and the challenge test.
[1] official score and 95% confidence interval computed on the challenge test set for task 1
[2] official score computed on the challenge test set for task 2

Dataset	Method	C-index (internal test)	C-index[1]	specificity (average)[2]	recall (average)[2]
A0	DSM	0.63	0.68 [0.65-0.71]	0.85	0.25
	NMES-CI	0.67	0.69 [0.66-0.73]	0.85	0.22
	NMES-CS	0.67	0.70 [0.67-0.73]	0.85	0.23
	TACE-RF	0.64	0.69 [0.66-0.73]	-	-
A6	DSM	0.69	0.75 [0.72-0.78]	0.88	0.34
	NMES-CI	0.72	0.73 [0.70-0.76]	0.86	0.23
	NMES-CS	0.70	0.74 [0.72-0.77]	0.86	0.29
	TACE-RF	0.69	0.71 [0.69-0.74]	-	-
B0	DSM	0.70	0.71 [0.68-0.74]	0.86	0.28
	NMES-CI	0.70	0.72 [0.69-0.75]	0.84	0.20
	NMES-CS	0.69	0.72 [0.70-0.75]	0.86	0.27
	TACE-RF	0.69	0.72 [0.69-0.75]	-	-
B6	DSM	0.74	0.74 [0.71-0.76]	0.87	0.28
	NMES-CI	0.75	0.74 [0.72-0.76]	0.85	0.21
	NMES-CS	0.73	0.75 [0.72-0.77]	0.87	0.32
	TACE-RF	0.76	0.72 [0.69-0.74]	-	-
C0	DSM	0.70	0.71 [0.68-0.74]	0.86	0.26
	NMES-CI	0.67	0.71 [0.68-0.74]	0.84	0.18
	NMES-CS	0.66	0.71 [0.69-0.74]	0.85	0.23
C6	DSM	0.75	0.74 [0.71-0.76]	0.87	0.28
	NMES-CI	0.70	0.73 [0.70-0.76]	0.84	0.22
	NMES-CS	0.70	0.74 [0.72-0.77]	0.86	0.26

NMES models, which treat all events as if they were independent. This may be due to some peculiarity of the challenge dataset or to the handling of competing risks in the DSM method, which may be unable to take advantage of the multiple events it received in training.

6. Acknowledgments

This work was supported by the European Union's Horizon 2020 Brainteaser Project (GA101017598).

References

[1] A. Guazzo, I. Trescato, E. Longato, E. Hazizaj, D. Dosso, G. Faggioli, G. M. Di Nunzio, G. Silvello, M. Vettoretti, E. Tavazzi, C. Roversi, P. Fariselli, S. C. Madeira, M. de Carvalho, M. Gromicho, A. Chiò, U. Manera, A. Dagliati, G. Birolo, H. Aidos, B. Di Camillo, N. Ferro, Intelligent Disease Progression Prediction: Overview of iDPP@CLEF 2022, in:

A. Barrón-Cedeño, G. Da San Martino, M. Degli Esposti, F. Sebastiani, C. Macdonald, G. Pasi, A. Hanbury, M. Potthast, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Thirteenth International Conference of the CLEF Association (CLEF 2022), Lecture Notes in Computer Science (LNCS) 13390, Springer, Heidelberg, Germany, 2022.

[2] A. Guazzo, I. Trescato, E. Longato, E. Hazizaj, D. Dosso, G. Faggioli, G. M. Di Nunzio, G. Silvello, M. Vettoretti, E. Tavazzi, C. Roversi, P. Fariselli, S. C. Madeira, M. de Carvalho, M. Gromicho, A. Chiò, U. Manera, A. Dagliati, G. Birolo, H. Aidos, B. Di Camillo, N. Ferro, Overview of iDPP@CLEF 2022: The Intelligent Disease Progression Prediction Challenge, in: G. Faggioli, N. Ferro, A. Hanbury, M. Potthast (Eds.), CLEF 2022 Working Notes, CEUR Workshop Proceedings (CEUR-WS.org), ISSN 1613-0073, 2022.

[3] C. Nagpal, X. Li, A. Dubrawski, Deep survival machines: Fully parametric survival regression and representation learning for censored data with competing risks, IEEE Journal of Biomedical and Health Informatics 25 (2021) 3163–3175.

[4] S. Pölsterl, scikit-survival: A library for time-to-event analysis built on top of scikit-learn, Journal of Machine Learning Research 21 (2020) 1–6. URL: http://jmlr.org/papers/v21/20-729.html.

[5] G. Ridgeway, The state of boosting, Comp Sci Stat 31 (2001).

[6] T. Hothorn, P. Bühlmann, S. Dudoit, A. Molinaro, M. J. Van Der Laan, Survival ensembles, Biostatistics 7 (2005) 355–373. URL: https://doi.org/10.1093/biostatistics/kxj011. doi:10.1093/biostatistics/kxj011.

[7] R. K. Vinayak, R. Gilad-Bachrach, Dart: Dropouts meet multiple additive regression trees, in: Artificial Intelligence and Statistics, PMLR, 2015, pp. 489–497.

[8] H. Ishwaran, U. B. Kogalur, E. H. Blackstone, M. S. Lauer, Random survival forests, The Annals of Applied Statistics 2 (2008) 841 – 860. URL: https://doi.org/10.1214/08-AOAS169. doi:10.1214/08-AOAS169.