Multi-Event Survival Prediction for Amyotrophic Lateral Sclerosis

Corrado Pancotti , Giovanni Birolo, Tiziana Sanavia , Cesare Rollo, Piero Fariselli
@unito.it

Dept. of Medical Sciences, University of Turin

The iDDP challenge at a glance

Two input datasets:

- all data collected at diagnosis (time of first ALSFRS-R)
- all data collected up to six months after diagnosis (only for patients without events up to six months)

Three survival outcomes with the following events:

- first event between NIV and death
- first event between PEG and death
- only death

Two tasks:

- Task 1 or A. predict a risk score (C-index and time dependent AUC)
- Task 2 or B: predict a time-to-event

Data processing and Methods

Data preprocessing

General preprocessing:

- features with more than 90% of missing values were discarded,
- categorical and boolean features were one-hot-encoded,
- almost constant one-hot-encoded values (99%) were discarded,
- retained features were standardized (mean=0, std=1) and imputed with the mean.

Longitudinal features were present in the prediction at 6 months:

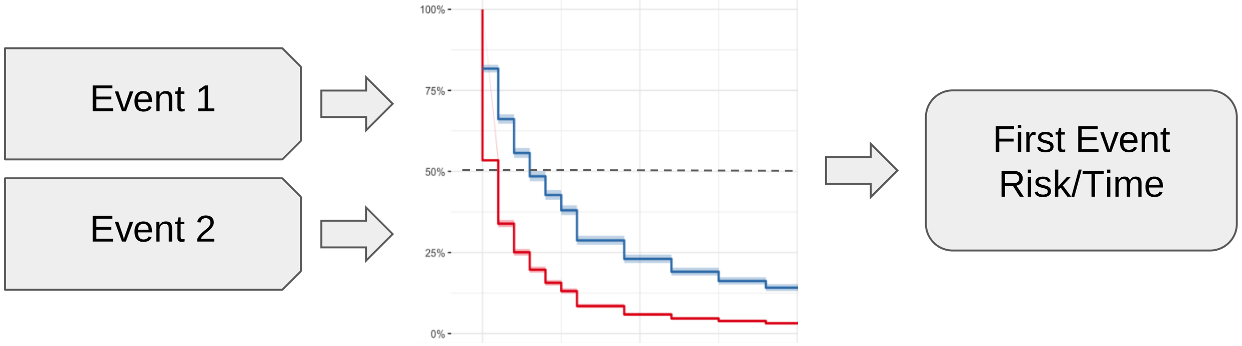
- 2-3 ALSFRS-R questionnaires and 0-2 FVC measurements per patient,
- the values of each longitudinal feature were encoded into 7 static features:

mean, max, min, first, last, standard deviation and slope.

Model 1: Naive Multiple Event Survival (NMES)

Basic Idea: handle competing risks as independent with standard models

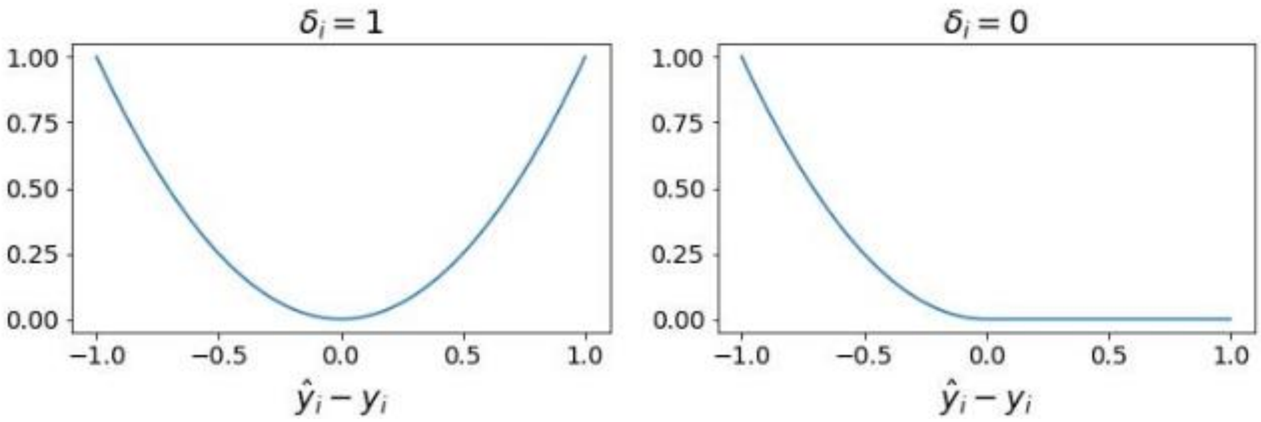
- one survival model for each event
- treat other events as censoring (naive)
- pick the event/model whose predicted survival curve reaches 0.5 first or whose time to event is shorter
- Tested Machine Learning Models: 1. Elastic Net Cox, 2. Random Survival Forest, 3. Gradient Boosting with regression trees



Model 1: Naive Multiple Event Survival (NMES)

Model Losses:

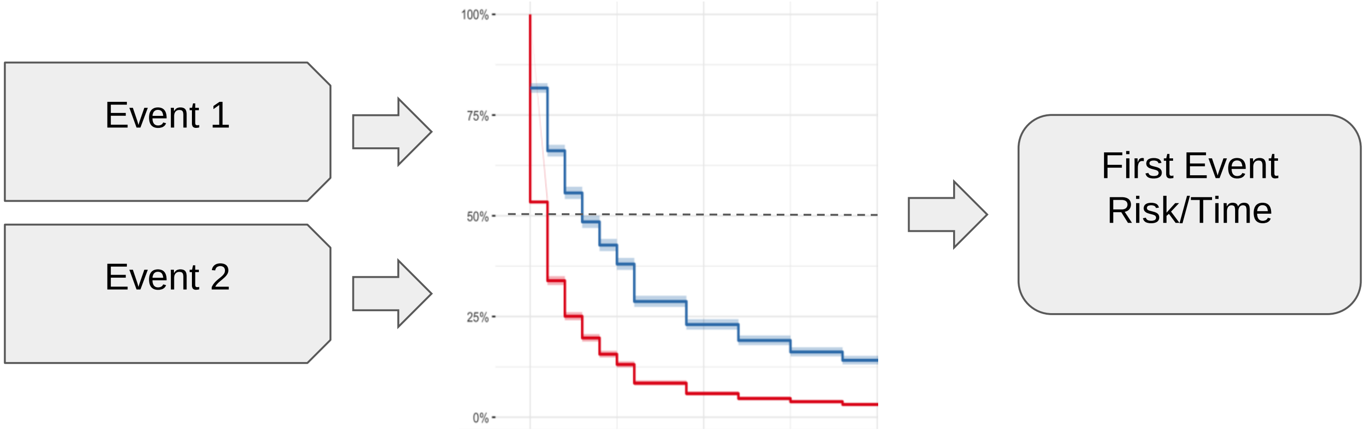
- Elastic Net Cox => penalized partial likelihood
- Random Survival Forest => maximal survival-time difference split
- Gradient Boosting with regression trees => Regression-based



Model 1: Naive Multiple Event Survival (NMES)

Hyper-parameter selection was performed twice with different metrics

- NMES-CI: model was selected only by its c-index
- NMES-CS: “combo” metrics, a weighted linear combination of c-index (60%), accuracy (40%)

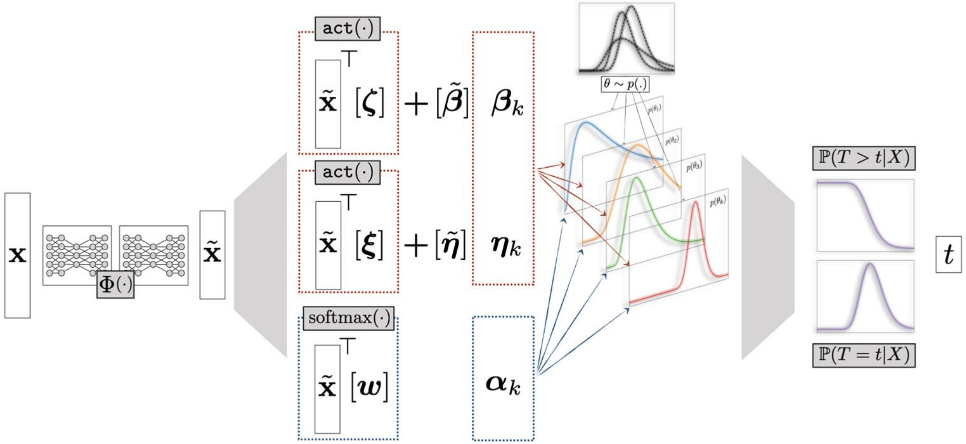


Model 2: Deep Survival Machines (DSM)*

Deep learning/statistical approach (Nagpal et al 2020*)

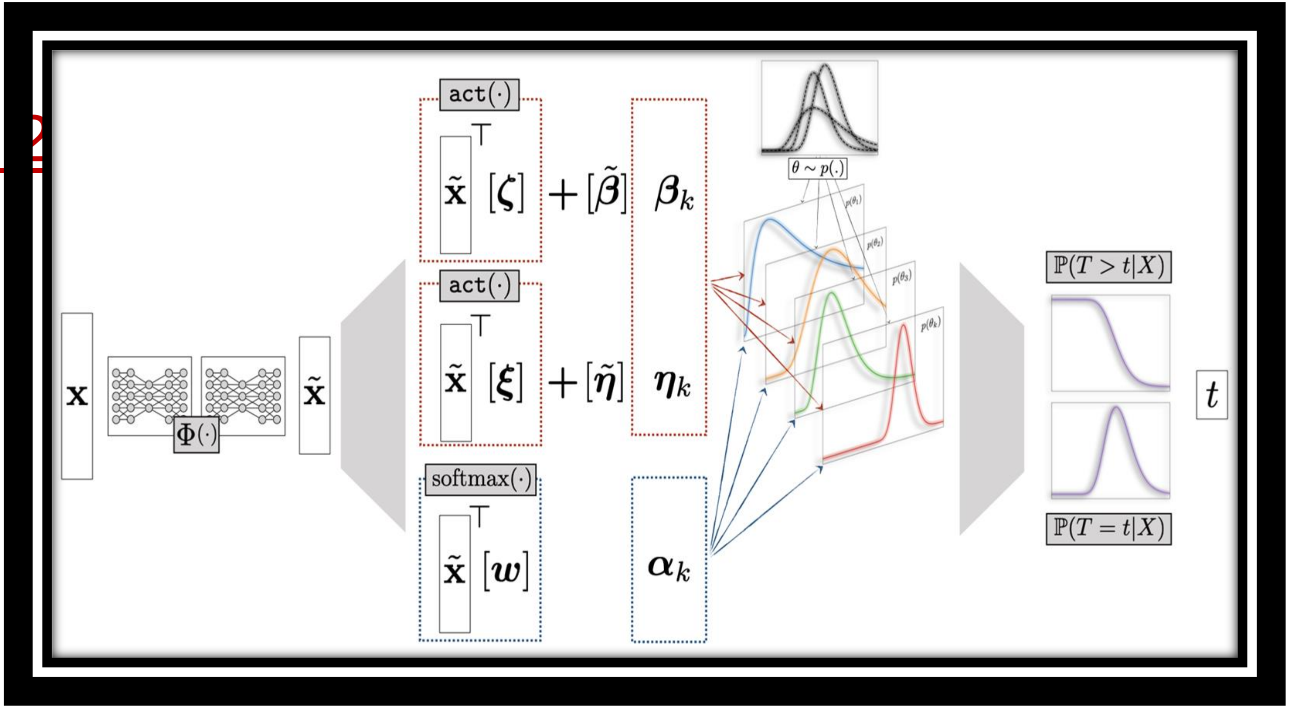
The neural network predicts parametric survival distributions

- event density modeled as mixture of LogNormal and Weibull
- distribution parameters are computed by a neural network
- handles competing risks by fitting all events separately (censoring others) but with weight sharing



* C.Nagpal, X.R. Li, A. Dubrawski 2020 https://arxiv.org/abs/2003.01176

Model 2



The input features, x are passed through a deep multilayer perceptron followed by a softmax over mixture size, K. The Conditional Distribution of P(T |X = x) is then described as a mixture of K (PRIMITIVE distributions, drawn from some prior.)

Model 3: Time Aware Classifier Ensemble (TACE)

IDEA: Create an ensemble of *N* classifiers by cutting the dataset at different random times. For each cut buind the *t*-th classifier.

For k in 1..N

cut the dataset at a random time t,

take events before t as positive, drop censored events before t

events and censored after t as negative

build the t-th classifier (Random Forest was only tested)

Compute the risk as the mean score across the ensemble: $\bar{y} = \sum_{t=1}^{N} y_t / N$

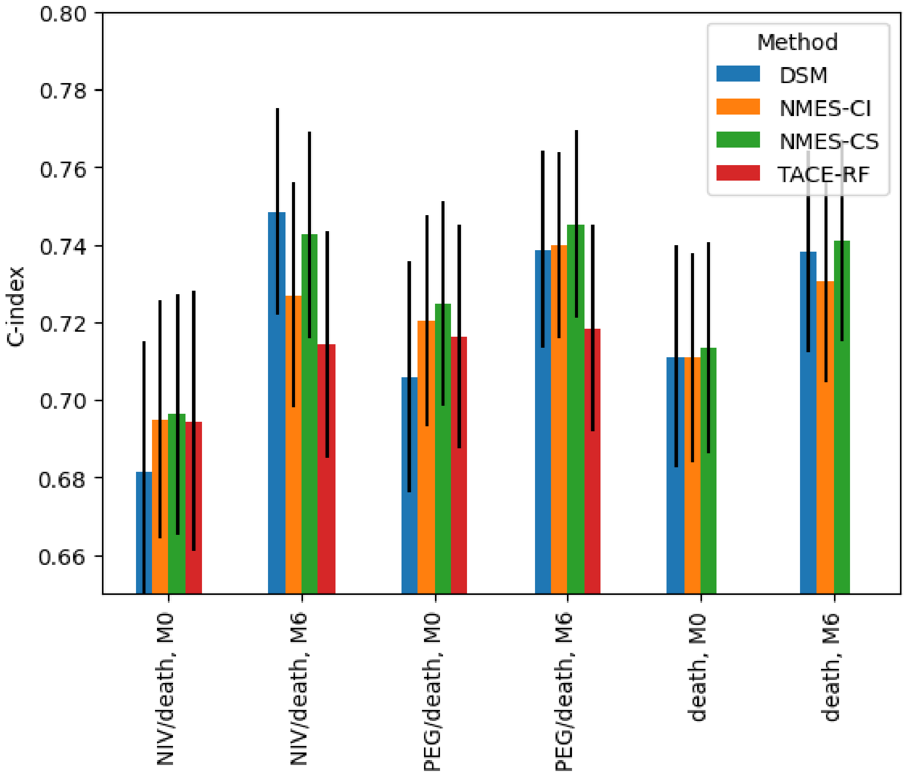
this model does not predict time-to-event so we did not submit it for Task 2

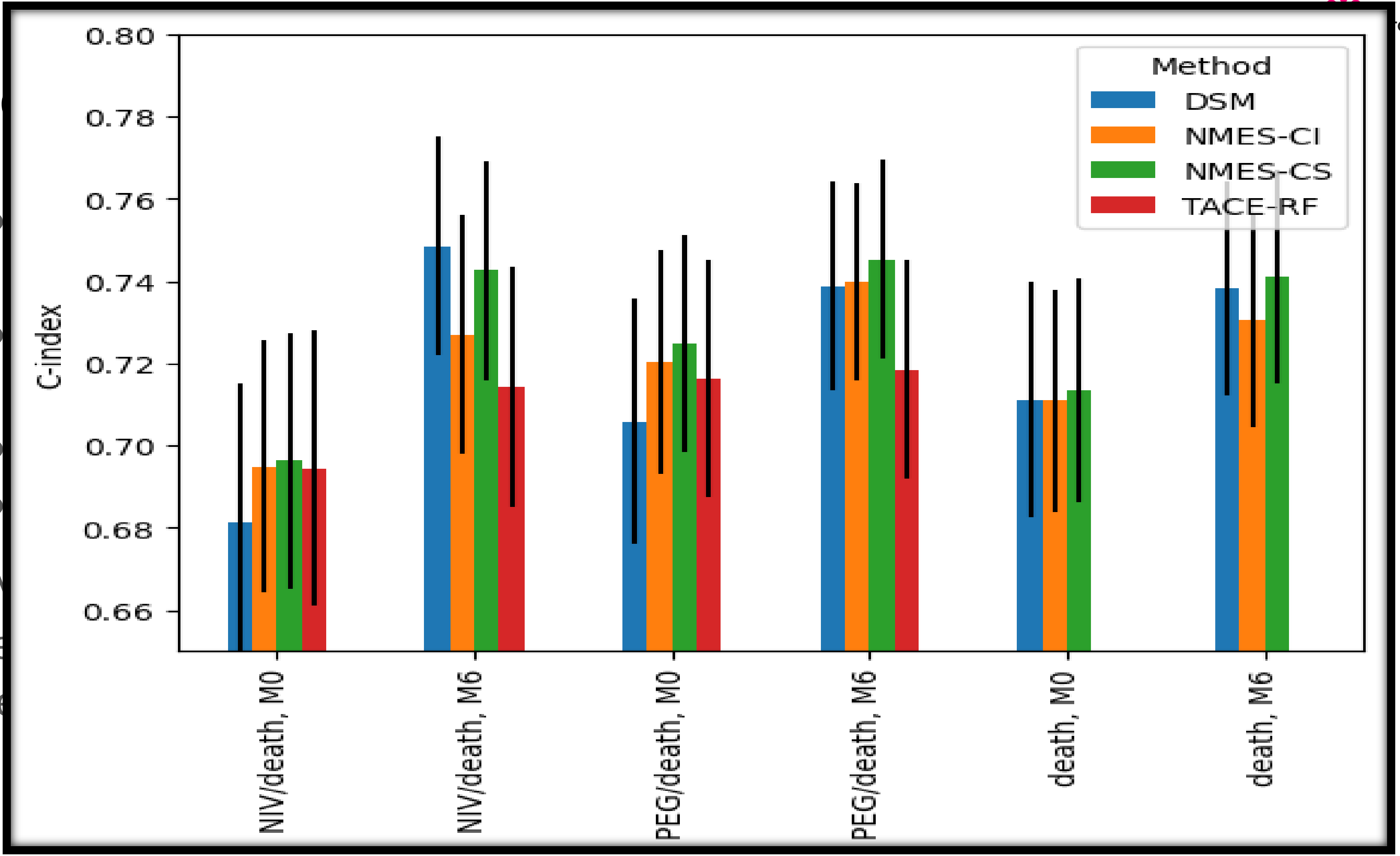
Results

Performance Overview (C-index)

- DSM: weaker at diagnosis (M0), better at M6
- NMES-CS: consistently high performance
- NMES-CI: close to NMES-CS
- TACE-RF: weaker at M6, better at M0

NMES always selected Gradient Boosting Regression Trees as the best underlying method in optimization.




Method
DSM
NMES-CI
NMES-CS
TACE-RF
C-index
0.80
0.78
0.76
0.74
0.72
0.70
0.68
0.66
NIV/death, M0
NIV/death, M6
PEG/death, M0
PEG/death, M6
death, M0
death, M6

Performance for NIV/DEATH (Task A)

Dataset	Method	C-index (internal)	C-index (challenge)	Specificity (challenge)	Recall (challenge)
Diagnosis NIV/DEATH	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]	-	-
Six months NIV/DEATH	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]	-	-

Performance for PEG/DEATH (Task B)

Dataset	Method	C-index (internal)	C-index (challenge)	Specificity (challenge)	Recall (challenge)
Diagnosis PEG/DEATH	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]**	-	-
Six months PEG/DEATH	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]	-	-

Performance for DEATH (Task C)

Dataset	Method	C-index (internal)	C-index (challenge)	Specificity (challenge)	Recall (challenge)
Diagnosis DEATH	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
Six months DEATH	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

Conclusions

- In general, better prediction with six-month information.
- DSM and NMES were close in performance
- TACE performed slightly worse (however, ensembles were small), Indicating that survival framework is better suited that classification ones.
- Apparently treating events as independent or competing makes little difference, even though they are not actually independent (death prevents NIV and PEG)
- Multi-state survival models should be tested

Thank you for the attention !

Brainteaser