ML for Survival Analysis and Dynamic Survival Analysis

davide.morelli@eng.ox.ac.uk

Goals

- How to approach survival analysis (time-to-event)
  - Preprocessing
  - ML
  - DL
- Longitudinal covariates (time series)
  - How this changes the setting
  - Techniques
- A bit of historical overview
- Domain specific loss functions

Summary

ML for Survival Analysis:

- Boosting
- Survival Random Forests
- DeepSurv

Dynamic Survival Analysis:

- Motivation: longitudinal data / trajectories
- Time dependent concordance Index
- Methods for dynamic survival analysis: the old days
    - 2 stage approach, joint models
- Methods for dynamic survival analysis: today
    - Deep Learning
        - MatchNet
        - RNN-Surv
        - Dynamic-deephit

Davide Morelli

Academia

- PhD Computer Science @ Pisa
- Lecturer @ Pisa
- Research and Teaching Fellow @ Surrey
- DPhil candidate Biomedical Engineering @ Ox

Industry

- CTO @ BioBeats
- ML Lead @ Huma Therapeutics

ML for Survival Analysis

penalised Cox regression models

boosted survival models

random survival forests

Feature Selection Method

Machine Learning Algorithm	(1) None	(2) Univariate	(3) RF Var Imp	(4) RF Min Depth	(5) RF Max Stat	(5) mRMR	(7) SFS	(8) SFFS
CoxPH (1)	0.612	0.808	0.815	0.824	0.803	0.511	0.787	0.783
Ridge (2)	0.8	0.809	0.812	0.815	0.815	0.5		
ElasticNet (3)	0.805	0.815	0.816	0.819	0.818	0.5		
Lasso (4)	0.807	0.814	0.817	0.819	0.815	0.565		
CoxBoost (5)	0.82	0.817	0.818	0.828	0.817	0.546		
GLMBoost (6)	0.791	0.789	0.791	0.789	0.787	0.559	0.759	0.769
XGBoost-Tree (7)	0.792	0.686	0.706	0.718	0.706	0.514	0.697	0.694
XGBoost - Linear (8)	0.799	0.807	0.8	0.808	0.795	0.58	0.776	0.79
RF-Survival (9)	0.797	0.792	0.799	0.793	0.8	0.575	0.707	0.712
RF-MaxStat (10)	0.812	0.811	0.806	0.813	0.806	0.585	0.739	0.743

1.0 0.9 0.8 0.7 0.6 0.5

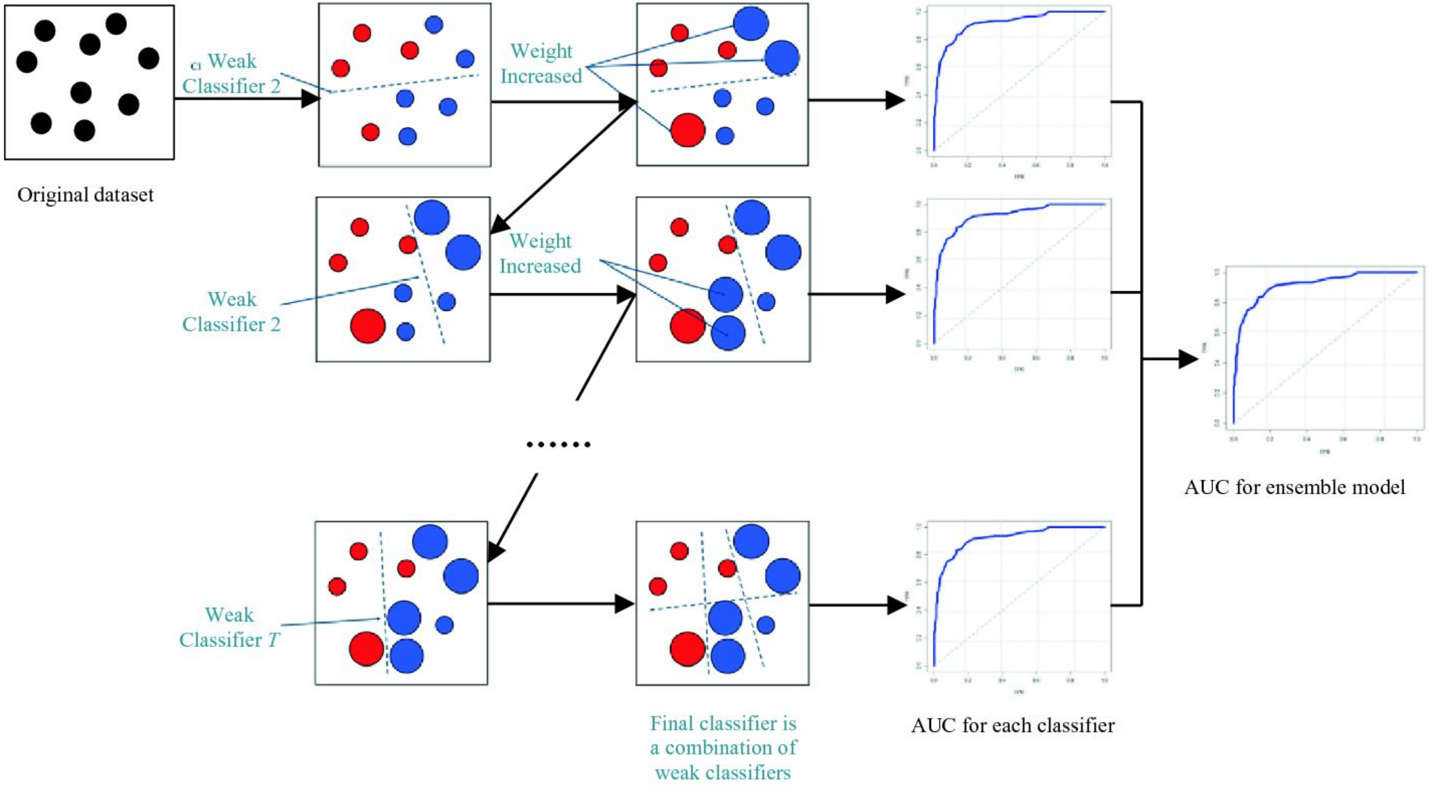
From [A comparison of machine learning methods for survival analysis of high-dimensional clinical data for dementia prediction]

Feature selection

- Filter Methods:
  - Univariate, Recursive feature elimination
  - Random forest variable importance
  - Random forest minimal depth
  - Maximally Selected Rank Statistics
  - MRMR: Maximum Relevance — Minimum Redundancy
- Wrapper Method:
  - Sequential forward selection
- Also:
  - Surrogate minimal depth
  - Permutation feature importance
  - Drop Column feature importance

See Supplementary Materials of "A comparison of machine learning methods for survival analysis of high-dimensional clinical data for dementia prediction"

ML for Survival Analysis: Boosting



Picture shamelessly stolen from [this blog post]

ML for Survival Analysis: Boosting

Boosting:

- Combine weak learners
- Sequential: next weak learner will "focus on the mistakes" of the current model
- The final model uses all weak learners

Models:

- XGBoost, CoxBoost

Still a active research area:

- E.g.: A combination of XGBoost, Cox, and ElasticNet: Wang, 2021

For differences between XGBoost and CoxBoost see De Bin, 2016

ML for Survival Analysis: Random Survival Forests

- See Random survival forests original paper (Ishwaran et al., 2008)
- Modification of Random Forest.
  - Draw B bootstraps of data
  - Grow a tree from each bootstrap
  - For every node select p random features, split using the feature that maximises difference in cumulative hazard function (CHF)
- CHF is Nelson–Aalen estimator

Deaths

$$\hat{H}_h(t) = \sum_{t_{l,h} \le t} \frac{d_{l,h}}{Y_{l,h}}.$$

Hazard at time t

Total subjects

DL for Survival Analysis

To fit CoxPH you minimise the partial log likelihood:

$$L(\beta, X) = -\sum_{i\in U}\left(\beta^T X_i - \log\sum_{j\in\Omega_i} e^{\beta^T X_j}\right)$$

For each subject i who had an event

Hazard ratio of patient i

Hazard ratio of every subject who had event after i or no event

Where’s the baseline hazard function?

From [Survival Analysis for Deep Learning]

DL for Survival Analysis

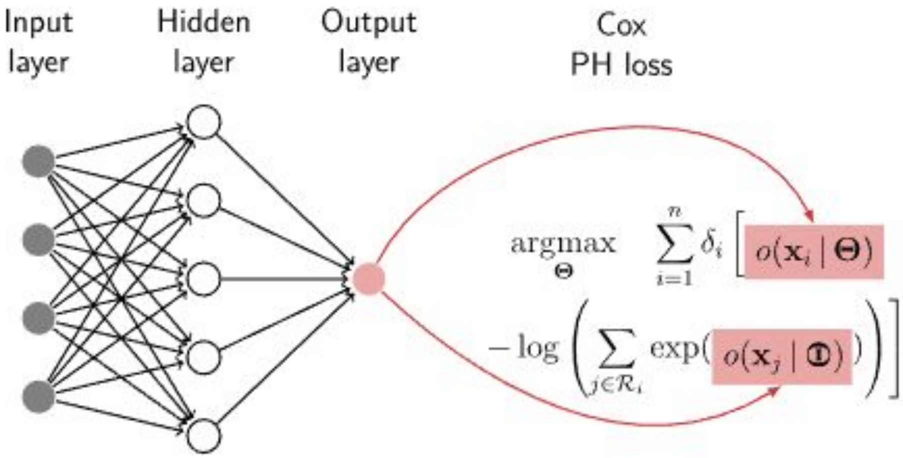
$$L(\beta, X) = -\sum_{i \in U}\left(\boxed{\beta^T X_i} - \log \sum_{j \in \Omega_i} e^{\boxed{\beta^T X_j}}\right)$$

They don’t **have** to be linear...

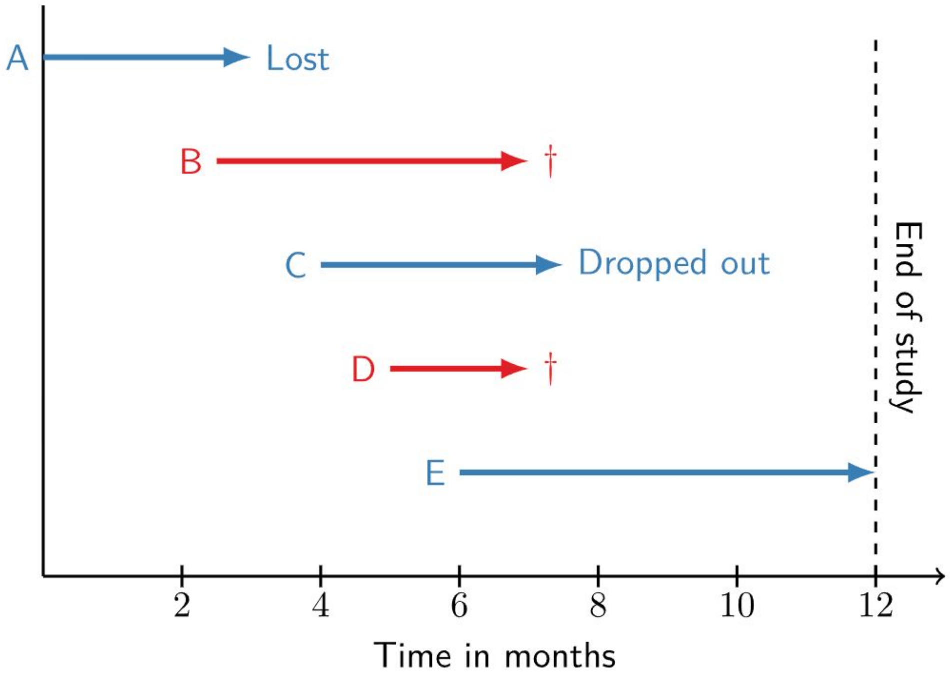
DL for Survival Analysis

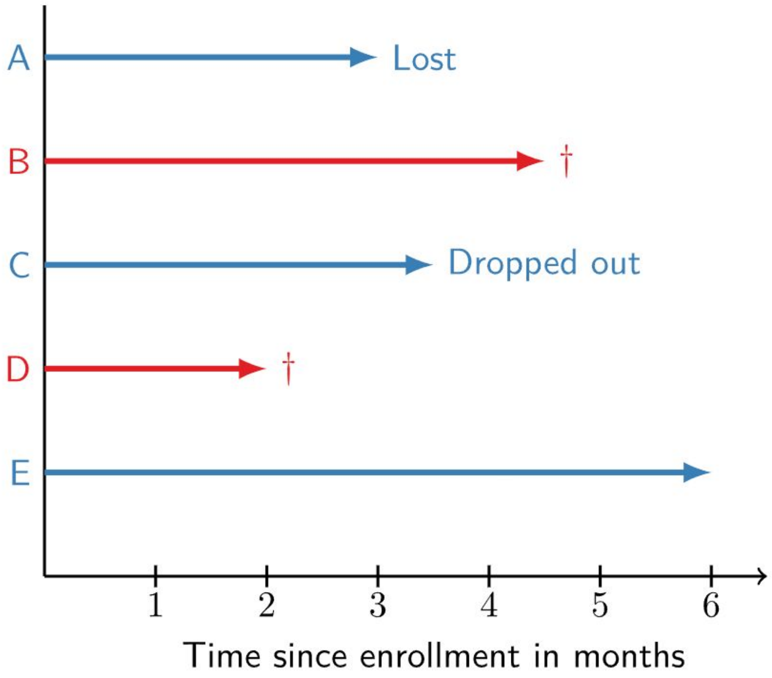
Deep Surv:

- Idea originally proposed in the work of Faraggi and Simon back in 1995
- Used in many papers since then: example1, example2, etc…
- PyTorch implementation



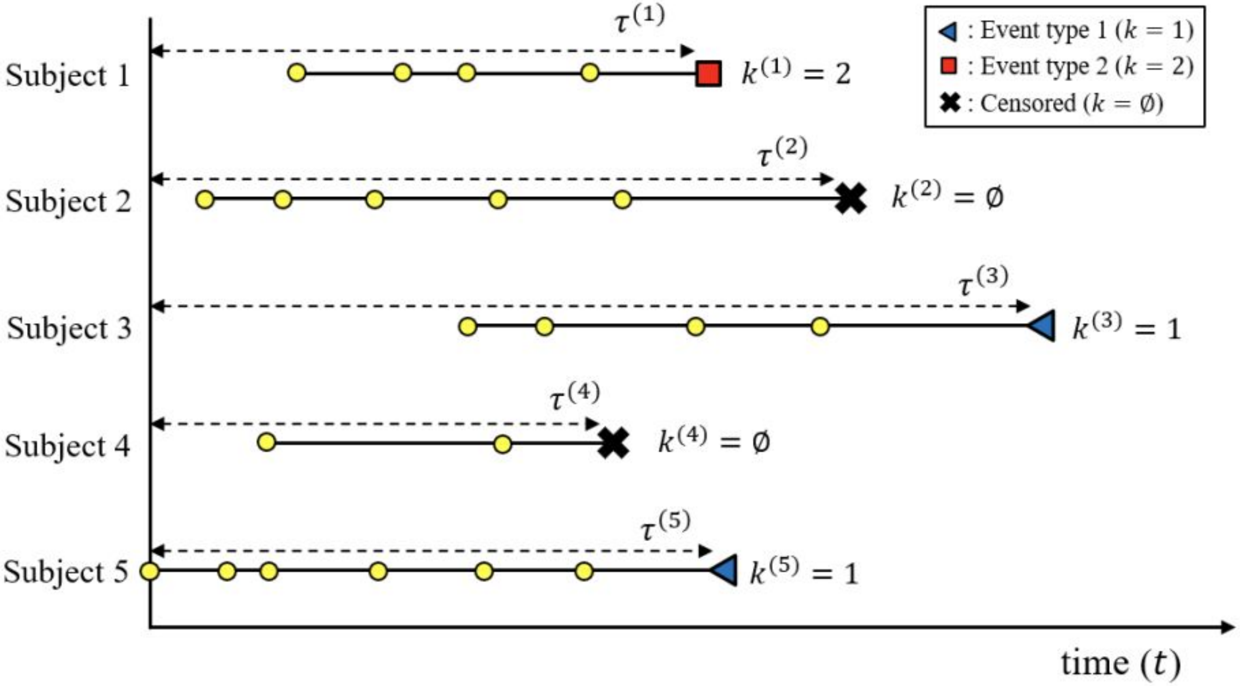
Static vs dynamic survival analysis





Picture taken from scikit-survival documentation

Static vs **dynamic** survival analysis



Picture taken from dynamic-deephit paper

Metric

C(t) : Time Dependent Concordance Index (CI)

- Proposed in Estimating a time-dependent concordance index for survival prediction models with covariate dependent censoring (2012)

Metric

C(t) : Time Dependent Concordance Index (CI)

$$\mathcal{C}_t(t) = E_{ij}\left\{1\{M_n(t, X_i) > M_n(t, X_j)\} \mid T_i < T_j, T_i \leqslant t, D_n^{\text{train}}\right\}$$

CI is now a function of t

Consider only pairs where subject i had an event before time t, and time of event for subject j is > i

Metric

C(t) : Time Dependent Concordance Index (CI)

$$\mathcal{C}_t(t) = E_{ij}\left\{1\{M_n(t, X_i) > M_n(t, X_j)\} \mid T_i < T_j, T_i \leqslant t, D_n^{\text{train}}\right\}$$

Indicator function: is 1 if risk of i is higher than risk of j

Metric

C(t) : Time Dependent Concordance Index (CI)

$$\mathcal{C}_t(t) = E_{ij}\left\{1\{M_n(t, X_i) > M_n(t, X_j)\} \mid T_i < T_j, T_i \leqslant t, D_n^{\mathrm{train}}\right\}$$

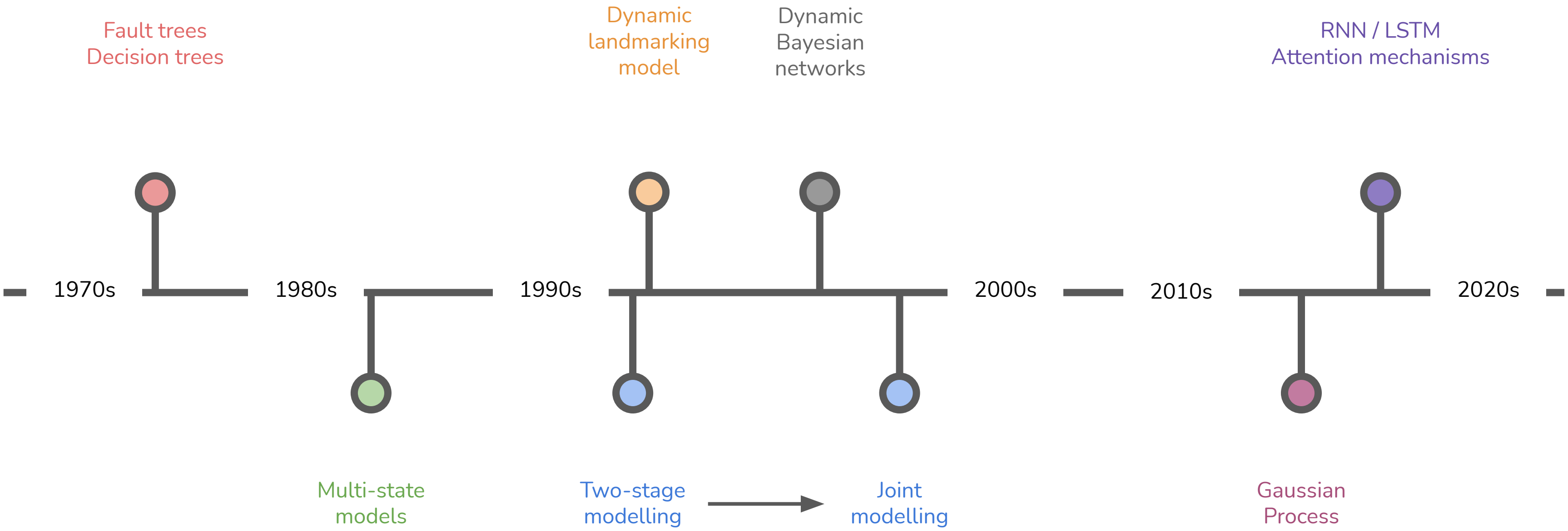
Expected value

```python
### C(t)-INDEX CALCULATION
def c_index(Prediction, Time_survival, Death, Time):
    '''
    N = len(Prediction)
    A = np.zeros((N,N))
    Q = np.zeros((N,N))
    N_t = np.zeros((N,N))
    Num = 0
    Den = 0
    for i in range(N):
        A[i, np.where(Time_survival[i] < Time_survival)] = 1
        Q[i, np.where(Prediction[i] > Prediction)] = 1

        if (Time_survival[i]<=Time and Death[i]==1):
            N_t[i,:] = 1

    Num  = np.sum(((A)*N_t)*Q)
    Den  = np.sum((A)*N_t)

    if Num == 0 and Den == 0:
        result = -1 # not able to compute c-index!
    else:
        result = float(Num/Den)

    return result
```

```
This is a cause-specific c(t)-index
- Prediction      : risk at Time (higher --> more risky)
- Time_survival   : survival/censoring time
- Death           :
    > 1: death
    > 0: censored (including death from other cause)
- Time            : time of evaluation (time-horizon when evaluating C-index)
```

From dynamic-deephit source code

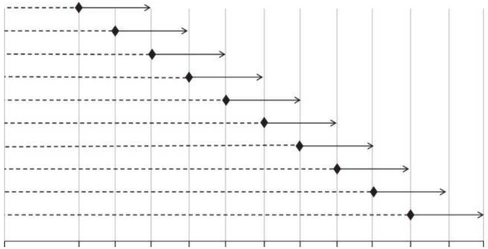
Evolution of approaches to dynamic risk analysis



Evolution of approaches to dynamic risk analysis

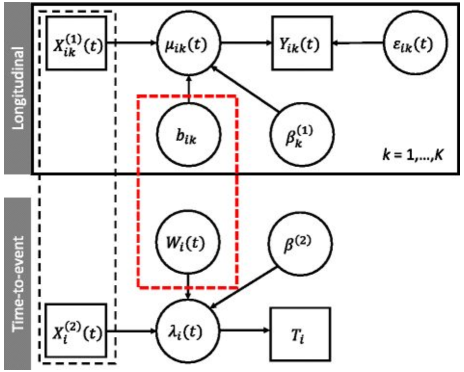
Landmark models

- Timescale is divided into a discrete set of landmark times
- Separate model is trained at each landmark time for those who are still at risk of the event, as the horizon of the prediction moves forward



Two-stage modelling

- First, longitudinal model is fitted (usually a linear mixed model)
- Next, time-to-event model (Cox PH) is fitted using the parameters of the longitudinal model



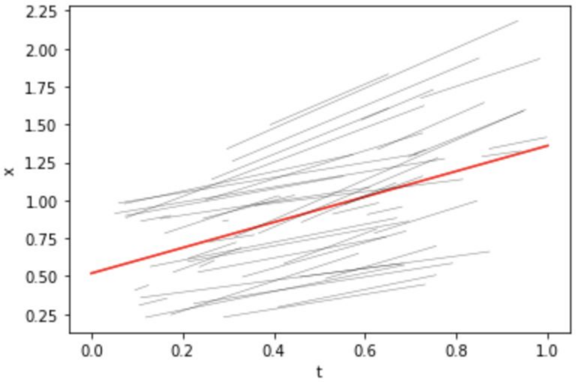
Joint modelling

- Requires model for the survival outcome, model for the dynamic marker process and method to link them (often shared random effects)
- Defining joint distribution of the two associated processes
- Can be computationally demanding

Layla Parast and Tianxi Cai. Landmark risk prediction of residual life for breast cancer survival. Statistics in medicine, 32(20):3459–3471, 2013
Hickey, G.L., Philipson, P., Jorgensen, A. *et al.* Joint modelling of time-to-event and multivariate longitudinal outcomes: recent developments and issues. *BMC Med Res Methodol* 16, 117 (2016). https://doi.org/10.1186/s12874-016-0212-5
Michael J Sweeting and Simon G Thompson. Joint modelling of longitudinal and time-to-event data with application to predicting abdominal aortic aneurysm growth and rupture. Biometrical Journal, 53 (5):750–763, 2011

2 stage approach: stage 1 - linear mixed model

Stage 1:



Example: each vital sign has a “typical trajectory” over time (e.g. blood pressure increases), i.e. the red line

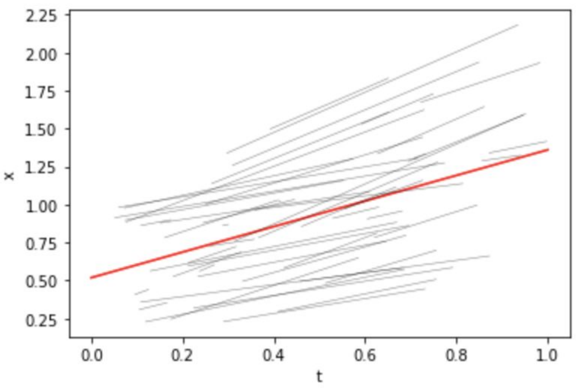
2 stage approach: stage 1 - linear mixed model

fixed effect parameters

error at time t for subject i

measure at time i for subject i

Stage 1: $\boxed{x_i(t)} = \boxed{\beta_0} + \quad \boxed{(\beta_1} \quad)t + \boxed{\beta_2}\text{age}_i + \boxed{\epsilon_i(t)}$



Example: each vital sign has a “typical trajectory” over time (e.g. blood pressure increases), i.e. the red line

2 stage approach: stage 1 - linear mixed model

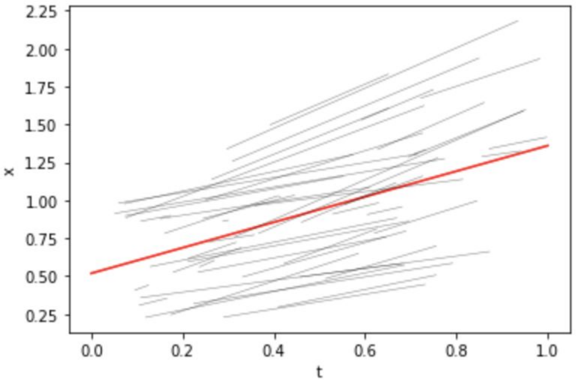
fixed effect parameters

error at time t for subject i

measure at time i for subject i

Stage 1: $$\boxed{x_i(t)} = \boxed{\beta_0} + \boxed{b_{0,i}} + (\boxed{\beta_1} + \boxed{b_{1,i}})t + \boxed{\beta_2}\text{age}_i + \boxed{\epsilon_i(t)}$$

patient specific intercept and slope



Example: each vital sign has a “typical trajectory” over time (e.g. blood pressure increases), i.e. the red line

We build a parametric model of how each subject deviates from that typical trajectory, i.e. the grey lines

2 stage approach: stage 2 - regression

fixed effect parameters

error at time t for subject i

measure at time i for subject i

Stage 1: $$\boxed{x_i(t)} = \boxed{\beta_0} + \boxed{b_{0,i}} + (\boxed{\beta_1} + \boxed{b_{1,i}})t + \boxed{\beta_2}\text{age}_i + \boxed{\epsilon_i(t)}$$

patient specific intercept and slope

Stage 2: (logistic regression) $$y_i = \sigma(\alpha_0 + \alpha_1 \boxed{b_{0,i}} + \alpha_2 \boxed{b_{1,i}} + \alpha_3 \text{age}_i)$$

We use parameters of “deviation from normal trajectory” to inform a logistic regression model (or cox).

Play with the [interactive version]

Joint Model

fixed effect parameters

error at time t for subject i

measure at time i for subject i

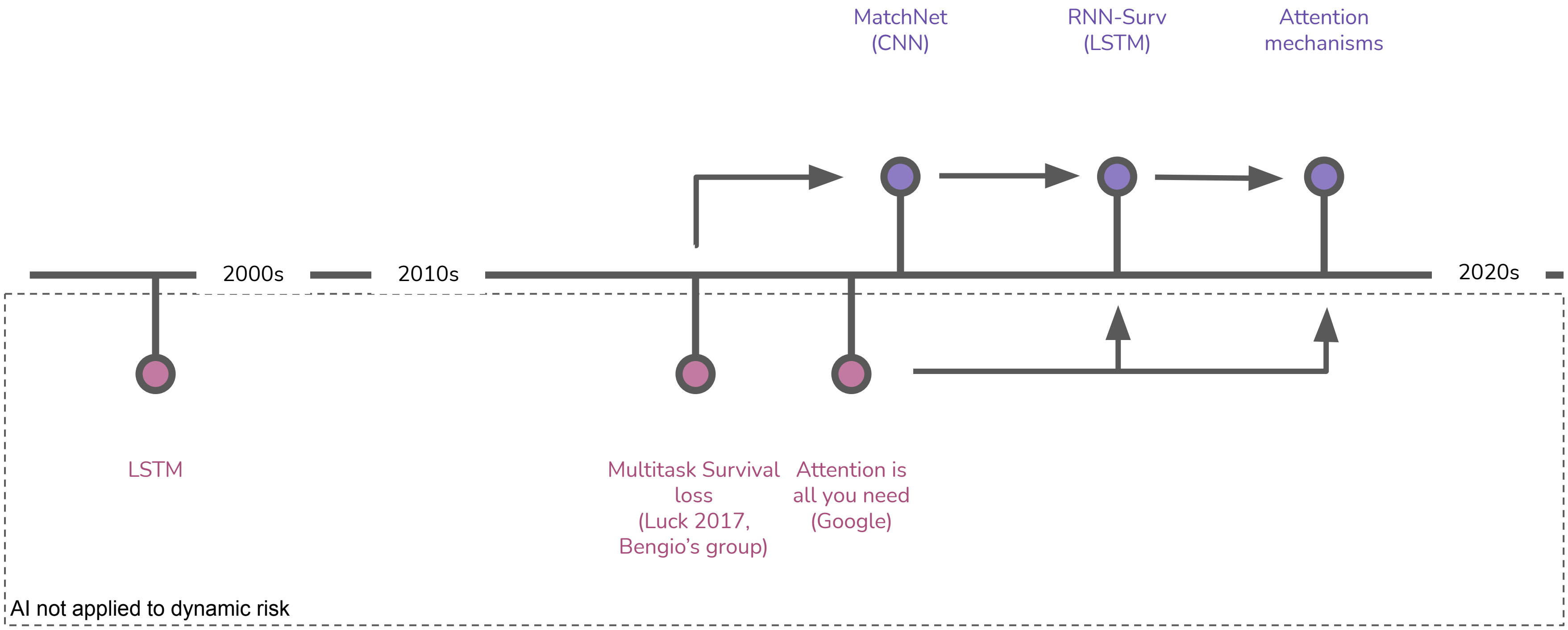
$$\boxed{x_i(t)} = \boxed{\beta_0} + \boxed{b_{0,i}} + \boxed{(\beta_1} + \boxed{b_{1,i}})t + \boxed{\beta_2}\text{age}_i + \boxed{\epsilon_i(t)}$$

Basically the same thing,
but the 2 stages are trained
simultaneously
(Monte Carlo Expectation Maximization)

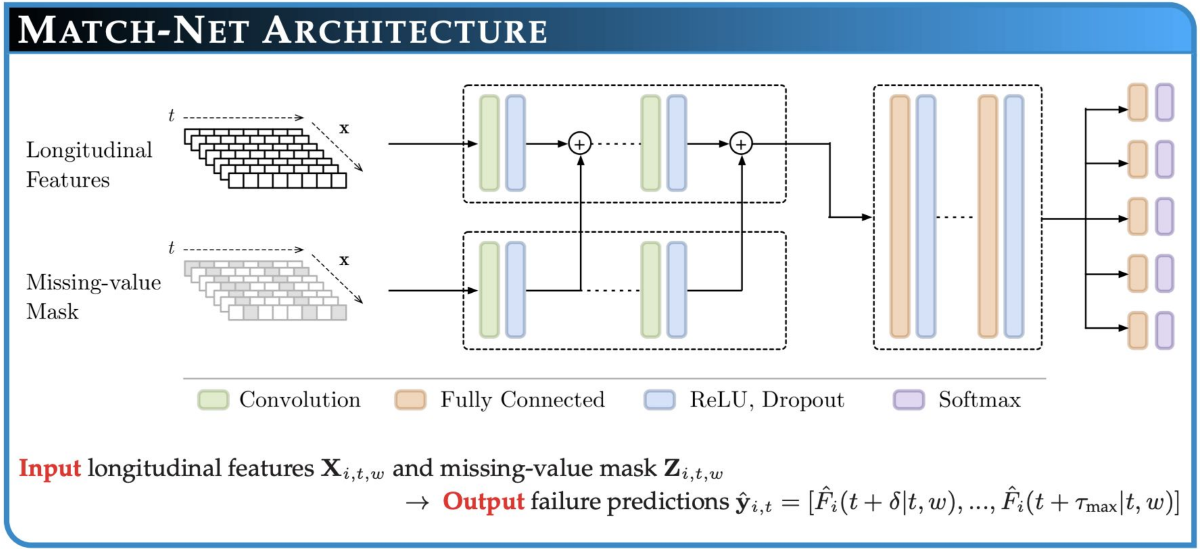
patient specific intercept and slope

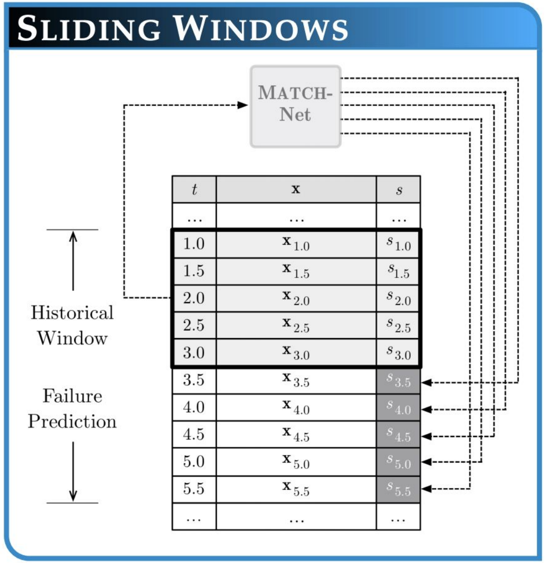
$$y_i = \sigma(\alpha_0 + \alpha_1 \boxed{b_{0,i}} + \alpha_2 \boxed{b_{1,i}} + \alpha_3 \text{age}_i)$$

Evolution of approaches to dynamic risk analysis: DL



Evolution of approaches to dynamic risk analysis: MatchNet





MATCH-Net: Dynamic Prediction in Survival Analysis using Convolutional Neural Networks Jarrett, 2018

Evolution of approaches to dynamic risk analysis: MatchNet

We want to estimate the failure function F:

$$F_i(t+\tau|t,w) = \mathbb{P}(T_{i,\text{surv}} \leq t+\tau | T_{i,\text{surv}} > t, \mathbf{X}_{i,t,w})$$

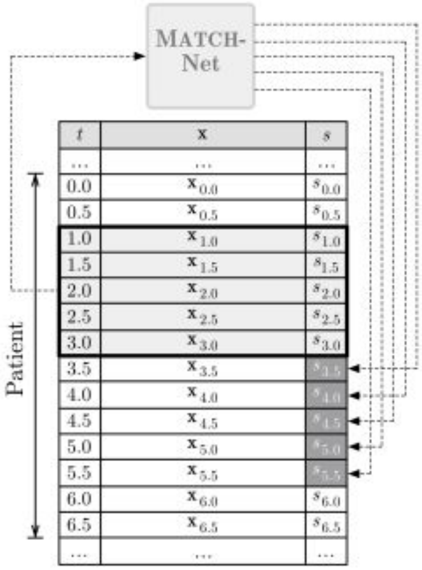
Probability that failure will happen before t + tau, given that:

- it hasn't happened at time t yet
- The predictors are known at time t

Evolution of approaches to dynamic risk analysis: MatchNet

The output layer gives us failure estimates for a window of timepoints:

$$\hat{\mathbf{y}}_{i,t} = [\hat{F}_i(t+\delta|t,w), \ldots, \hat{F}_i(t+\tau_{\max}|t,w)]$$



(c) Sliding windows

Evolution of approaches to dynamic risk analysis: MatchNet

The total loss:

$$l(\boldsymbol{\theta}) = \frac{\delta^2}{\sum_{i=1}^{N} \sum_{j=1}^{t_i} \tau_i} \sum_{i=1}^{N} \sum_{j=1}^{t_i/\delta} \sum_{k=1}^{\tau_i/\delta} \alpha(i, (j-1)\delta, k\delta) \cdot l_{i,(j-1)\delta,k\delta}$$

For all subjects i

Evolution of approaches to dynamic risk analysis: MatchNet

The total loss:

$$l(\boldsymbol{\theta}) = \frac{\delta^2}{\sum_{i=1}^{N} \sum_{j=1}^{t_i} \tau_i} \sum_{i=1}^{N} \sum_{j=1}^{t_i/\delta} \sum_{k=1}^{\tau_i/\delta} \alpha(i, (j-1)\delta, k\delta) \cdot l_{i,(j-1)\delta,k\delta}$$

For all times j

Evolution of approaches to dynamic risk analysis: MatchNet

The total loss:

$$l(\boldsymbol{\theta}) = \frac{\delta^2}{\sum_{i=1}^{N} \sum_{j=1}^{t_i} \tau_i} \sum_{i=1}^{N} \sum_{j=1}^{t_i/\delta} \sum_{k=1}^{\tau_i/\delta} \alpha(i, (j-1)\delta, k\delta) \cdot l_{i,(j-1)\delta,k\delta}$$

For all prediction horizons k

Evolution of approaches to dynamic risk analysis: MatchNet

DEPARTMENT OF ENGINEERING SCIENCE

UNIVERSITY OF OXFORD

The total loss:

$$l(\boldsymbol{\theta}) = \frac{\delta^2}{\sum_{i=1}^{N}\sum_{j=1}^{t_i}\tau_i}\sum_{i=1}^{N}\sum_{j=1}^{t_i/\delta}\sum_{k=1}^{\tau_i/\delta}\alpha(i,(j-1)\delta,k\delta)\cdot l_{i,(j-1)\delta,k\delta}$$

Every subject has a weight that depends on survival duration

Evolution of approaches to dynamic risk analysis: MatchNet

The total loss:

$$l(\boldsymbol{\theta}) = \frac{\delta^2}{\sum_{i=1}^{N} \sum_{j=1}^{t_i} \tau_i} \sum_{i=1}^{N} \sum_{j=1}^{t_i/\delta} \sum_{k=1}^{\tau_i/\delta} \alpha(i, (j-1)\delta, k\delta) \cdot l_{i,(j-1)\delta,k\delta}$$

The negative log likelihood
of a single estimate

Evolution of approaches to dynamic risk analysis: MatchNet

DEPARTMENT OF ENGINEERING SCIENCE
UNIVERSITY OF OXFORD

the negative log likelihood for a single estimate is just binary cross entropy:

y log(p) + (1-y) log(1-p)

$$l_{i,t,\tau}(\boldsymbol{\theta}) = -[s_{i,t+\tau} \log \hat{F}_i(t+\tau|t,w) + (1-s_{i,t+\tau}) \log(1-\hat{F}_i(t+\tau|t,w))]$$

F is the model estimate

s is the binary survival indicator

Evolution of approaches to dynamic risk analysis: MatchNet

	τ	MATCH-Net	S-TCN	S-MLP	FCN	D-Atlas	RNN	MLP	JM	LM
AUROC	0.5	**0.962**	0.961	0.959	0.954	0.959	0.949*	0.948*	0.913*	0.909*
	1.0	**0.942**	0.941	0.932	0.930	0.929	0.930	0.930	0.917*	0.914*
	1.5	**0.902**	**0.902**	0.897	0.895	0.892	0.891	0.890	0.881	0.878
	2.0	**0.909**	0.908	0.904	0.903	0.896	0.901	0.895	0.894	0.890
	2.5	**0.886**	0.884	0.881	0.883	0.884	0.883	0.874	0.883	0.878
AUPRC	0.5	**0.594**	0.580	0.500	0.536	0.517	0.464*	0.469*	0.473*	0.469*
	1.0	**0.513**	0.505	0.447	0.453	0.423	0.410*	0.435	0.415*	0.412*
	1.5	**0.373**	0.367	0.354	0.357	0.364	0.340	0.340	0.319	0.325
	2.0	**0.390**	0.380	0.364	0.375	0.352	0.355	0.359	0.362	0.367
	2.5	**0.384**	0.381	0.371	0.365	0.360	0.365	0.356	0.366	0.363

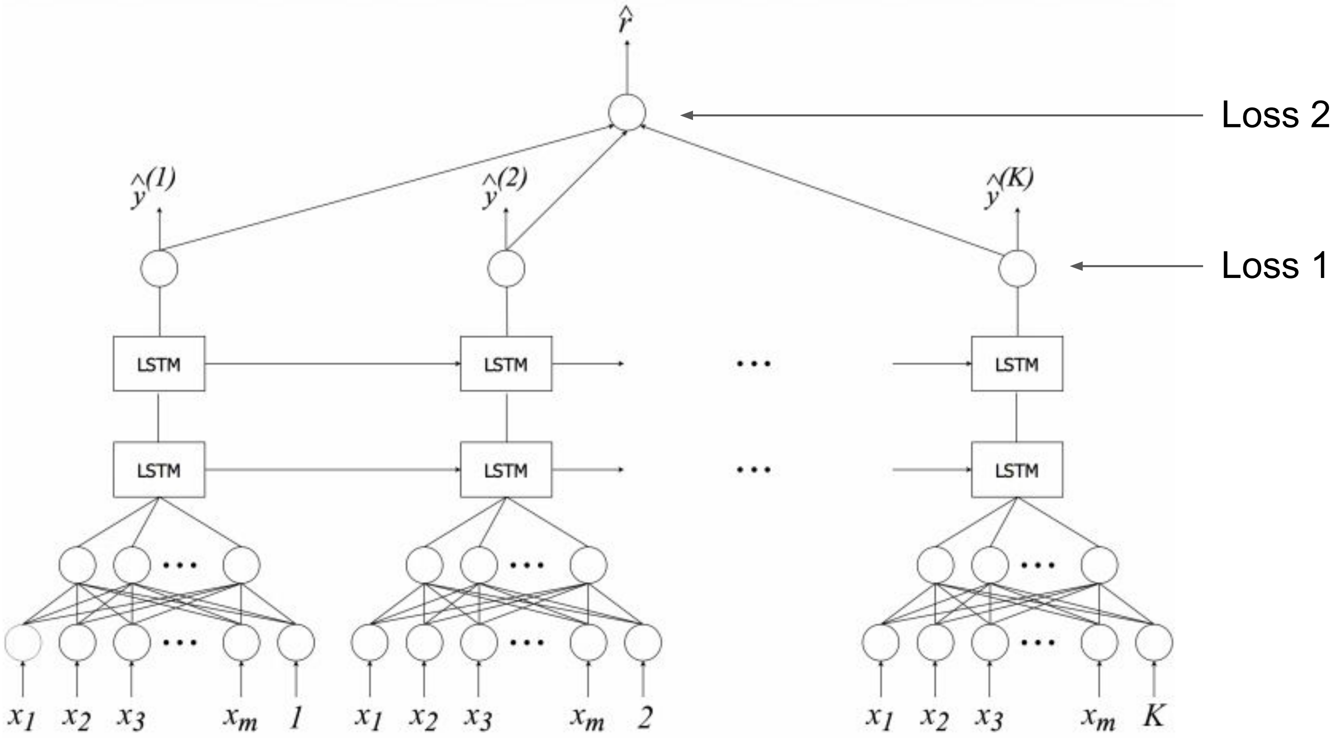
MatchNet without clinical input

MLP instead of CNN

Fully connected CNN

Disease Atlas

Evolution of approaches to dynamic risk analysis: RNN-Surv



RNN-SURV: A Deep Recurrent Model for Survival Analysis Giunchiglia, 2018

Evolution of approaches to dynamic risk analysis: RNN-Surv

Multitask learning

$$\mathcal{L}_1 = -\sum_{k=1}^{K} \sum_{i \in U_k} \left[\mathbb{I}(Y_i > t_k) \log \hat{y}_i^{(k)} + (1 - \mathbb{I}(Y_i > t_k)) \log(1 - \hat{y}_i^{(k)}) \right]$$

Cross entropy to predict time to event, modified to support censoring

For each time

Evolution of approaches to dynamic risk analysis: RNN-Surv

Multitask learning

$$\mathcal{L}_1 = -\sum_{k=1}^{K} \sum_{i \in U_k} \left[\mathbb{I}(Y_i > t_k) \log \hat{y}_i^{(k)} + (1 - \mathbb{I}(Y_i > t_k)) \log(1 - \hat{y}_i^{(k)}) \right]$$

Cross entropy to predict time to event, modified to support censoring

For each admissible subject

Evolution of approaches to dynamic risk analysis: RNN-Surv

Multitask learning

$$\mathcal{L}_1 = -\sum_{k=1}^{K} \sum_{i \in U_k} \left[\mathbb{I}(Y_i > t_k) \log \hat{y}_i^{(k)} + (1 - \mathbb{I}(Y_i > t_k)) \log(1 - \hat{y}_i^{(k)}) \right]$$

Cross entropy to predict time to event, modified to support censoring

Binary cross entropy

Evolution of approaches to dynamic risk analysis: RNN-Surv

Multitask learning

Maximise, for every admissible couple i and j, where Ti < Tj

$$\mathcal{L}_2 = -\frac{1}{|\mathcal{C}|} \sum_{(i,j)\in\mathcal{C}} \left[1 + \left(\frac{\log \sigma(\hat{r}_j - \hat{r}_i)}{\log 2}\right)\right]$$

Evolution of approaches to dynamic risk analysis: RNN-Surv

$$\sigma(z) = 1/(1+e^{-z})$$

Sigmoid

$$\mathbf{1}_{z>0} \geq 1 + (\log \sigma(z)/\log 2)$$

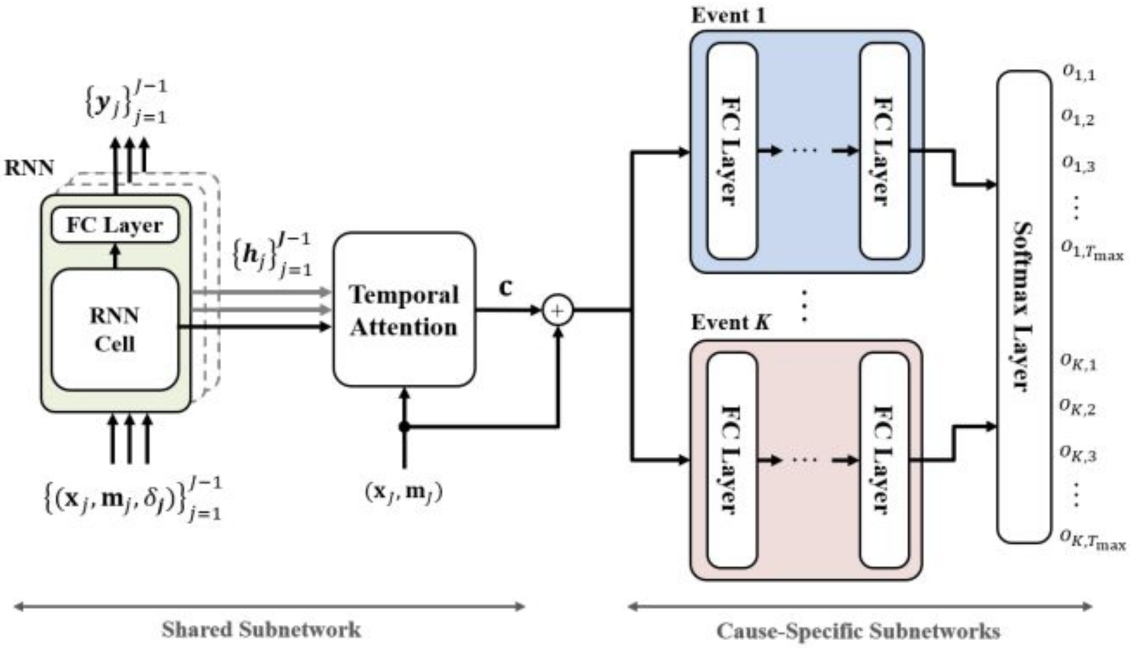
Lower bound to the indicator function

$$\mathcal{L}_2 = -\frac{1}{|\mathcal{C}|} \sum_{(i,j)\in\mathcal{C}} \left[1 + \left(\frac{\log \sigma(\hat{r}_j - \hat{r}_i)}{\log 2}\right)\right]$$

Upper bound to negative c-index,
see this paper

Play with the interactive version

Evolution of approaches to dynamic risk analysis: Dynamic-DeepHit



(a) The network architecture with K competing risks.

Dynamic-DeepHit: A Deep Learning Approach for Dynamic Survival Analysis With Competing Risks Based on Longitudinal Data Lee, 2020

Evolution of approaches to dynamic risk analysis: Dynamic-DeepHit

$$\mathcal{L}_{\text{total}} = \mathcal{L}_1 + \mathcal{L}_2 + \mathcal{L}_3$$

Log likelihood loss

Prediction loss

Ranking loss

Evolution of approaches to dynamic risk analysis: Dynamic-DeepHit

$$\mathcal{L}_{\text{total}} = \mathcal{L}_1 + \mathcal{L}_2 + \mathcal{L}_3$$

$\mathcal{L}_1 =$

Log likelihood loss:
If not censored: captures the event, and the time of the event.
If censored: captures the time of censoring

Evolution of approaches to dynamic risk analysis: Dynamic-DeepHit

$$\mathcal{L}_{\text{total}} = \mathcal{L}_1 + \mathcal{L}_2 + \mathcal{L}_3$$

$$\mathcal{L}_1 = -\sum_{i=1}^{N} \left[\mathbb{1}(k^i \neq \varnothing) \cdot \log\left(\frac{o^i_{k^i,\tau^i}}{1 - \sum_{k\neq\varnothing} \sum_{n \leq t^i_{J^i}} o^i_{k,n}} \right) + \mathbb{1}(k^i = \varnothing) \cdot \log\left(1 - \sum_{k\neq\varnothing} \hat{F}_k(\tau^i|\mathcal{X}^i) \right) \right]$$

For all subjects

We want to
maximise the sum

Evolution of approaches to dynamic risk analysis: Dynamic-DeepHit

$$\mathcal{L}_{\text{total}} = \mathcal{L}_1 + \mathcal{L}_2 + \mathcal{L}_3$$

$$\mathcal{L}_1 = -\sum_{i=1}^{N} \left[\mathbb{1}(k^i \neq \varnothing) \cdot \log\left(\frac{o^i_{k^i,\tau^i}}{1 - \sum_{k\neq\varnothing} \sum_{n \leq t^i_{J^i}} o^i_{k,n}} \right) + \mathbb{1}(k^i = \varnothing) \cdot \log\left(1 - \sum_{k\neq\varnothing} \hat{F}_k(\tau^i | \mathcal{X}^i) \right) \right]$$

o: output prob for risk k at time t
This is the Cumulative Incidence Function

Indicator function for risk k.
1 if event k happened

Evolution of approaches to dynamic risk analysis: Dynamic-DeepHit

$$\mathcal{L}_{\text{total}} = \mathcal{L}_1 + \mathcal{L}_2 + \mathcal{L}_3$$

$$\mathcal{L}_1 = -\sum_{i=1}^{N} \left[\mathbb{1}(k^i \neq \varnothing) \cdot \log\left(\frac{o^i_{k^i,\tau^i}}{1 - \sum_{k \neq \varnothing} \sum_{n \leq t^i_{J^i}} o^i_{k,n}} \right) + \mathbb{1}(k^i = \varnothing) \cdot \log\left(1 - \sum_{k \neq \varnothing} \hat{F}_k(\tau^i | \mathcal{X}^i) \right) \right]$$

Event didn't happen (censored subject)

1 - estimated Cumulative Incidence Function

Evolution of approaches to dynamic risk analysis: Dynamic-DeepHit

DEPARTMENT OF ENGINEERING SCIENCE
UNIVERSITY OF OXFORD

$$\mathcal{L}_{\text{total}} = \mathcal{L}_1 + \mathcal{L}_2 + \mathcal{L}_3$$

$\mathcal{L}_2 =$ Ranking loss: Cumulative Incidence Function at different times

Evolution of approaches to dynamic risk analysis: Dynamic-DeepHit

$$\mathcal{L}_{\text{total}} = \mathcal{L}_1 + \mathcal{L}_2 + \mathcal{L}_3$$

$\mathcal{L}_2 =$

Acceptable pairs: (i, j) where
i: experiences event k at time i
j: does not experience any event until i

We want our risk estimate for i larger than for j

Evolution of approaches to dynamic risk analysis: Dynamic-DeepHit

$$\mathcal{L}_{\text{total}} = \mathcal{L}_1 + \mathcal{L}_2 + \mathcal{L}_3$$

Hyperparameter for risk k

$$\mathcal{L}_2 = \sum_{k=1}^{K} \alpha_k \sum_{i \neq j} A_{kij} \cdot \eta\left(\hat{F}_k(s^i + t^i_{J^i}|\mathcal{X}^i), \hat{F}_k(s^i + t^j_{J^i}|\mathcal{X}^j)\right)$$

For all types of event

For all acceptable pairs (A is the indicator function for acceptable pairs). If the pair is not acceptable this becomes 0 and is ignored

Evolution of approaches to dynamic risk analysis: Dynamic-DeepHit

$$\mathcal{L}_{\text{total}} = \mathcal{L}_1 + \mathcal{L}_2 + \mathcal{L}_3$$

$$\mathcal{L}_2 = \sum_{k=1}^{K} \alpha_k \sum_{i \neq j} A_{kij} \cdot \eta\left(\hat{F}_k(s^i + t^i_{J^i} | \mathcal{X}^i), \hat{F}_k(s^i + t^j_{J^j} | \mathcal{X}^j)\right)$$

$$\eta(a, b) = \exp(-\tfrac{a-b}{\sigma})$$

Estimated Cumulative Incidence Functions for i and j

Differentiable loss function that grows exponentially if risk of j is > risk of i

Play with the [interactive version]

Evolution of approaches to dynamic risk analysis: Dynamic-DeepHit

$$\mathcal{L}_{\text{total}} = \mathcal{L}_1 + \mathcal{L}_2 + \mathcal{L}_3$$

$\mathcal{L}_3 =$ Prediction loss: step-ahead prediction of covariates

Evolution of approaches to dynamic risk analysis: Dynamic-DeepHit

$$\mathcal{L}_{\text{total}} = \mathcal{L}_1 + \mathcal{L}_2 + \mathcal{L}_3$$

$$\mathcal{L}_3 = \beta \cdot \sum_{i=1}^{N} \sum_{j=0}^{J^i-1} \sum_{d\in\mathcal{I}} (1 - m^i_{j+1,d}) \cdot \zeta(x^i_{j+1,d}, y^i_{j,d})$$

hyperparameter

For all subjects

Evolution of approaches to dynamic risk analysis: Dynamic-DeepHit

$$\mathcal{L}_{\text{total}} = \mathcal{L}_1 + \mathcal{L}_2 + \mathcal{L}_3$$

$$\mathcal{L}_3 = \beta \cdot \sum_{i=1}^{N} \sum_{j=0}^{J^i - 1} \sum_{d \in \mathcal{I}} (1 - m^i_{j+1,d}) \cdot \zeta(x^i_{j+1,d}, y^i_{j,d})$$

For all times

Evolution of approaches to dynamic risk analysis: Dynamic-DeepHit

$$\mathcal{L}_{\text{total}} = \mathcal{L}_1 + \mathcal{L}_2 + \mathcal{L}_3$$

$$\mathcal{L}_3 = \beta \cdot \sum_{i=1}^{N} \sum_{j=0}^{J^i-1} \sum_{d \in \mathcal{I}} (1 - m^i_{j+1,d}) \cdot \zeta(x^i_{j+1,d}, y^i_{j,d})$$

If next measure is missing this becomes 0 (therefore ignored)

Evolution of approaches to dynamic risk analysis: Dynamic-DeepHit

$$\mathcal{L}_{\text{total}} = \mathcal{L}_1 + \mathcal{L}_2 + \mathcal{L}_3$$

Measure at time j+1

Predicted measure at time j

$$\mathcal{L}_3 = \beta \cdot \sum_{i=1}^{N} \sum_{j=0}^{J^i - 1} \sum_{d \in \mathcal{I}} (1 - m^i_{j+1,d}) \cdot \zeta(x^i_{j+1,d}, y^i_{j,d})$$

If continuous: $\zeta(a, b) = |a - b|^2$

If binary: $\zeta(a, b) = -a \log b - (1 - a) \log(1 - b)$

Summary

Goals:

- How to approach survival analysis (time-to-event)
    - Preprocessing
    - ML
    - DL
- Handling longitudinal covariates (time series)
    - How this changes the setting
    - Techniques
- A bit of historical overview
- Domain specific loss functions

Lab

Preprocessing:

- Understand your data
- One-hot-encoding
- Cross validation
- Normalisation
- Mean-imputation

ML for Survival Analysis:

- CoxBoost
- Survival Random Forests
- Survival Support Vector Machines

Dynamic Survival Analysis:

- Add censoring to known loss
- Where’s the bug?
- Write hyper-parameter search code

Questions?

davide.morelli@eng.ox.ac.uk