Censored Regression

Using a Multi-Task approach

Philipp Ratz (UQAM)

Machine Learning Methods

And their Issues

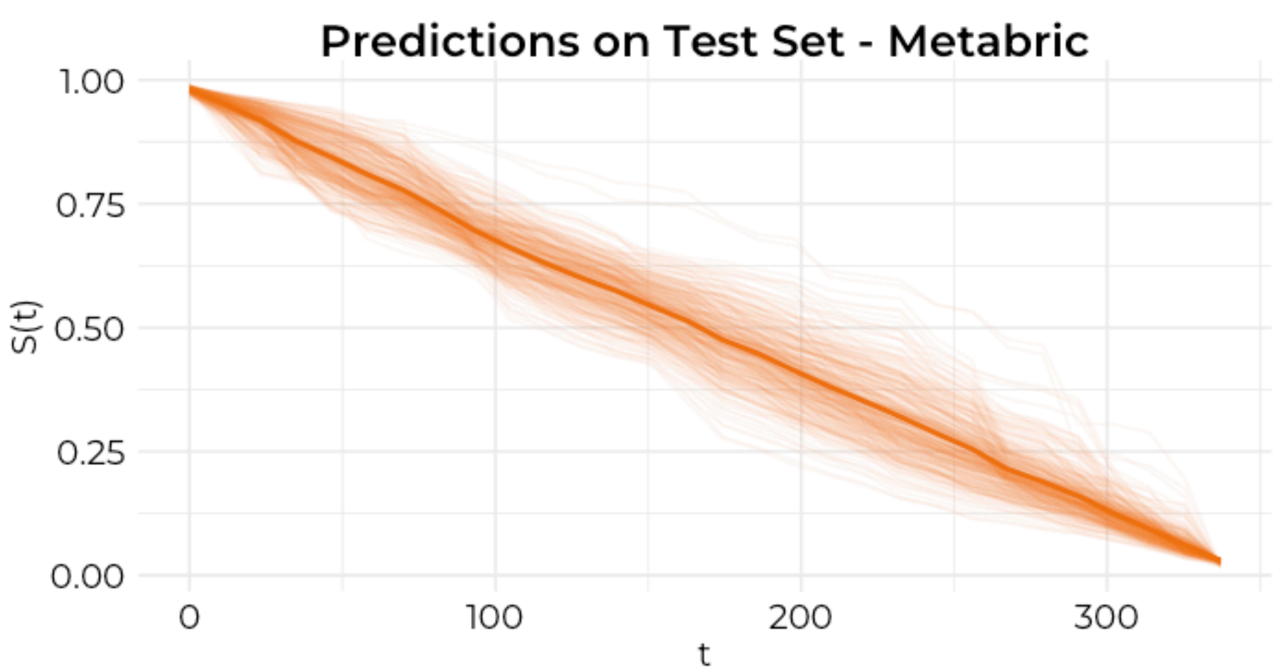
DeepHit (Lee et al. 2018) - still the benchmark to beat

Machine Learning Methods

And their Issues

DeepHit (Lee et al. 2018) - still the benchmark to beat

Excellent discriminative performance but what about calibration?
ie. $\mathbb{E}[Y \mid \hat{m}(\boldsymbol{X})] = \hat{m}(\boldsymbol{X})$

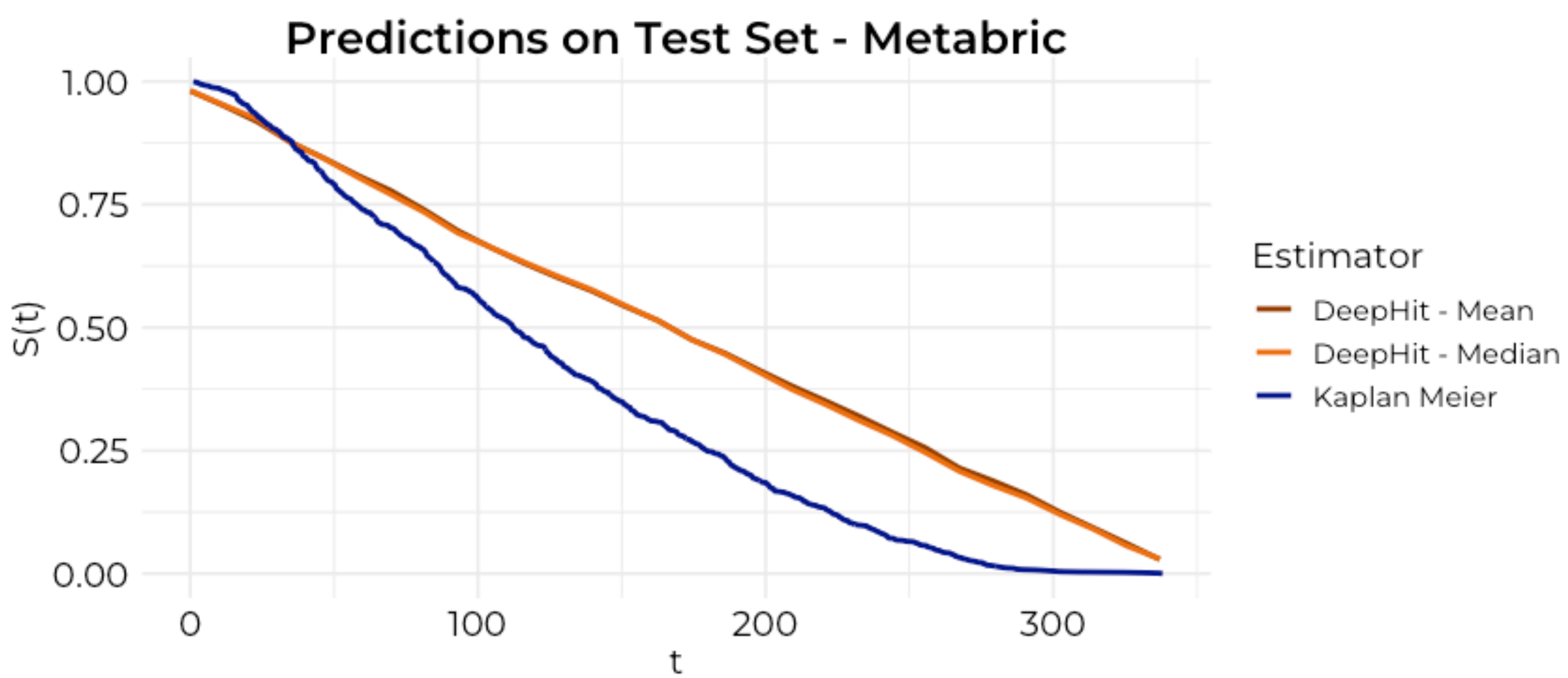


Machine Learning Methods

And their Issues

DeepHit (Lee et al. 2018) - still the benchmark to beat

Excellent discriminative performance but what about calibration?
ie. $\mathbb{E}[Y \mid \hat{m}(\boldsymbol{X})] = \hat{m}(\boldsymbol{X})$

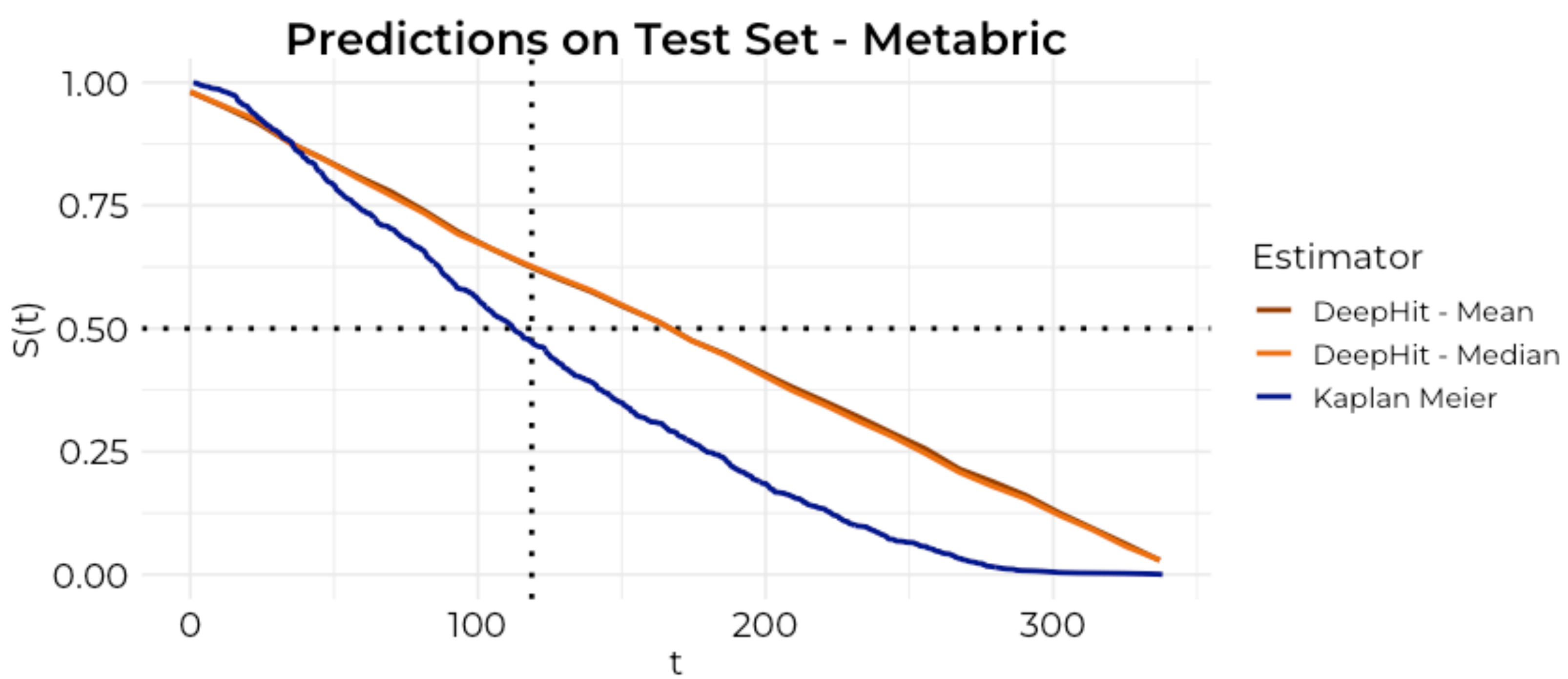


Machine Learning Methods

And their Issues

DeepHit (Lee et al. 2018) - still the benchmark to beat

Excellent discriminative performance but what about calibration?
ie. $\mathbb{E}[Y \mid \hat{m}(\boldsymbol{X})] = \hat{m}(\boldsymbol{X})$



Starting Point and Notation

- ML methods only recently started analysing the problem
- We are interested in a calibrated model
- The workhorse for many inference applications is still the Cox-PH model and its variants
- Today, spotlight on the simpler version - the *Kaplan Meier*

Starting Point and Notation

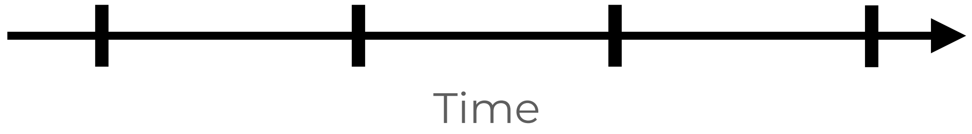
- ML methods only recently started analysing the problem
- We are interested in a calibrated model
- The workhorse for many inference applications is still the Cox-PH model and its variants
- Today, spotlight on the simpler version - the *Kaplan Meier*

Notation: We assume that for every individual we can observe the tuple $(\tilde{\tau}_i, \delta_i, x_i)$, where $\tilde{\tau}_i = \min\{\tau_i, c_i\}$ is the observation time, δ_i an indicator for censoring and x_i a vector of covariates. $S(t_j)$ denotes the survival function at time t_j and $h(t_j)$ the corresponding hazard rate.

Multi-Task Approach

- Pioneered by Yu et al. (2011). If we have tools to handle binary predictions, we can extend this to reformulate common survival problems

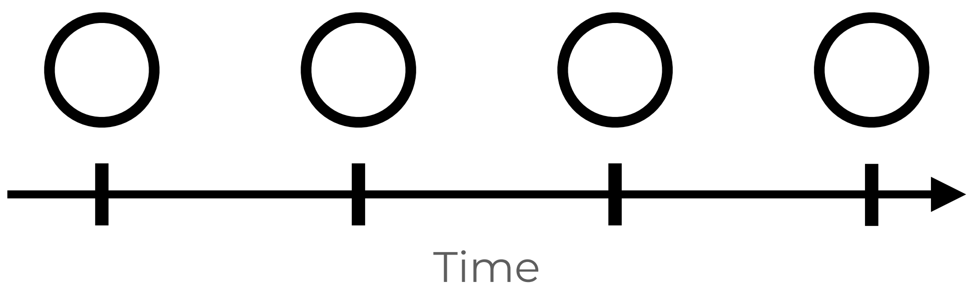
- Instead of directly modelling survival, consider a simple model for $z_j = \mathbb{P}(T \geq t_j | x)$



Multi-Task Approach

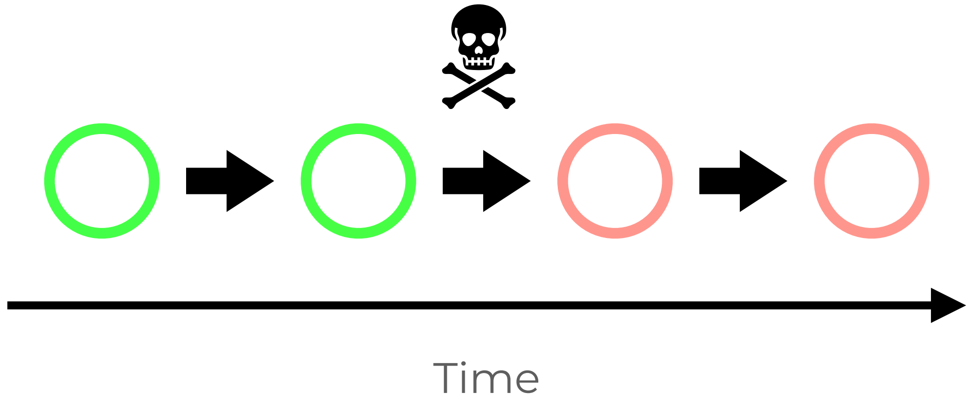
- Pioneered by Yu et al. (2011). If we have tools to handle binary predictions, we can extend this to reformulate common survival problems

- Instead of directly modelling survival, consider a simple model for $z_j = \mathbb{P}(T \geq t_j | x)$



Multi-Task Approach

- Pioneered by Yu et al. (2011). If we have tools to handle binary predictions, we can extend this to reformulate common survival problems
- Instead of directly modelling survival, consider a simple model for $z_j = \mathbb{P}(T \geq t_j \mid x)$
- But now construct a series of dependent regression tasks instead



(Conditioned) Kaplan-Meier

Setup - I

- Here, focus on the Kaplan-Meier estimator
- Easy to calculate the direction of estimation bias for a variety of scenarios
- Many procedures to correct for bias if it is known. See eg. Willems et al. (2018)

- Recall that the hazard rate and survival function are:

$$h(t_j) = \mathbb{P}[T = t_j | T \geq t_j] = \frac{p(t_j)}{S(t_{j-1})} \quad S(t_j) = 1 - \mathbb{P}[\tau = t_j | \tau \geq t_j] S(t_{j-1}) = \prod_{l=1}^{j} [1 - h(t_l)]$$

Conditioned Kaplan-Meier

Setup - II

- We need to consider censored instances

- Consider a weighting scheme creating a vector (or multi-task) estimation problem

$$Y_{i,j} = \begin{cases} 0 \text{ if } \tau_i < j \\ 1 \text{ otherwise} \end{cases} \forall i, j = 0,1,\ldots,K$$

$$W_{i,j} = \begin{cases} 0 \text{ if } c_i < j \\ 1 \text{ otherwise} \end{cases} \forall i, j = 0,1,\ldots,K$$

$$\tilde{\tau} = 4 \quad [1,1,1,1,0,\ldots,0]$$
$$c = 0 \quad [1,1,1,1,1\ldots,1]$$

$$\tilde{\tau} = 4 \quad [1,1,1,1,1,\ldots,1]$$
$$c = 1 \quad [1,1,1,1,0\ldots,0]$$

Conditioned Kaplan-Meier

Setup - II

- We need to consider censored instances

- Consider a weighting scheme creating a vector (or multi-task) estimation problem

$$Y_{i,j} = \begin{cases} 0 \text{ if } \tau_i > j \\ 1 \text{ otherwise} \end{cases} \forall i, j = 0,1,\ldots,K$$

$$W_{i,j} = \begin{cases} 0 \text{ if } c_i > j \\ 1 \text{ otherwise} \end{cases} \forall i, j = 0,1,\ldots,K$$

$$\tilde{\tau} = 4 \quad [1,1,1,1,0,\ldots,0]$$
$$c = 0 \quad [1,1,1,1,1\ldots,1]$$

$$\tilde{\tau} = 4 \quad [1,1,1,1,1,\ldots,1]$$
$$c = 1 \quad [1,1,1,1,0\ldots,0]$$

- Which yields the likelihood estimator(s)

$$\hat{z}_1 = \arg\max_{z_1} \prod_{i=1}^{n} z_1^{w_{i,1}y_{i,1}}(1 - z_1)^{w_{i,1}(1-y_{i,1})} \quad \cdots \quad \hat{z}_j = \arg\max_{z_j} \prod_{i=1}^{n} z_j^{w_{i,j}y_{i,j}}(1 - z_j)^{w_{i,j}(1-y_{i,j})}$$

Conditioned Kaplan-Meier

Setup - III

- Also need some restrictions (as in the original Kaplan-Meier estimation)

$$S(t_j) = 1 - \mathbb{P}[\tau = t_j \,|\, \tau \geq t_j] S(t_{j-1}) = \prod_{l=1}^{j} [1 - h(t_l)]$$

Conditioned Kaplan-Meier

Setup - III

- Also need some restrictions (as in the original Kaplan-Meier estimation)

$$S(t_j) = 1 - \mathbb{P}[\tau = t_j \,|\, \tau \geq t_j] S(t_{j-1}) = \prod_{l=1}^{j} [1 - h(t_l)]$$

- Here we simply impose directly:

$$\hat{z}_j = \begin{cases} \hat{q}(t_1) & \text{if } j = 1 \\ \hat{z}_{j-1}\hat{q}(t_j) & \text{if } j > 1 \end{cases}$$

Conditioned Kaplan-Meier

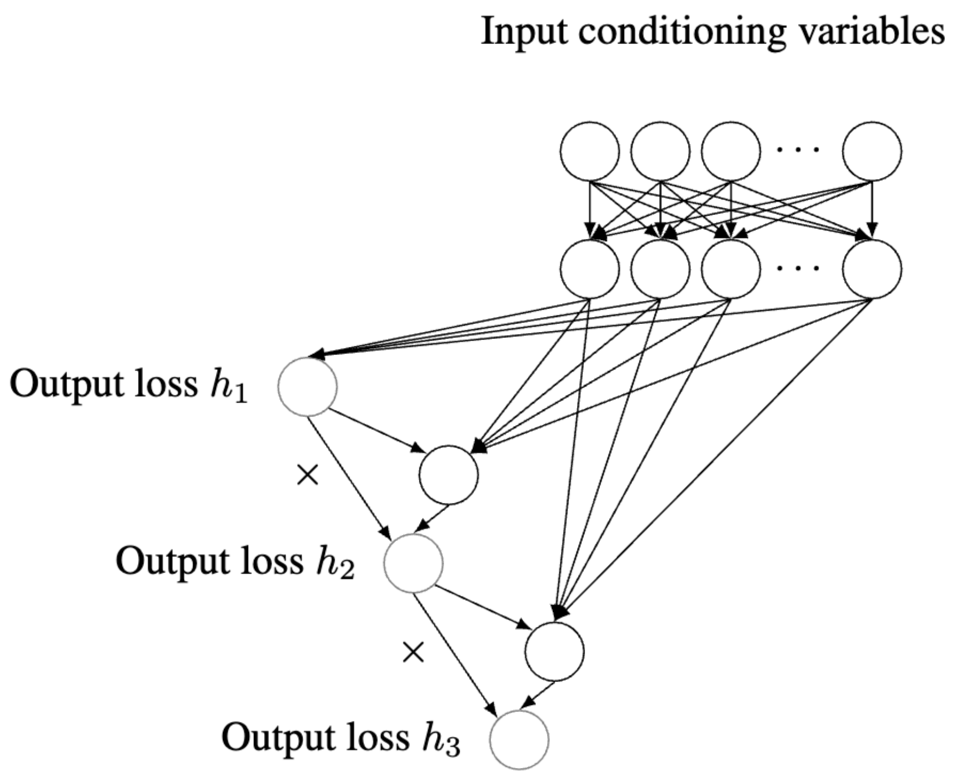
Setup - III

- Also need some restrictions (as in the original Kaplan-Meier estimation)

$$S(t_j) = 1 - \mathbb{P}[\tau = t_j \mid \tau \geq t_j] S(t_{j-1}) = \prod_{l=1}^{j} [1 - h(t_l)]$$

- Here we simply impose directly:

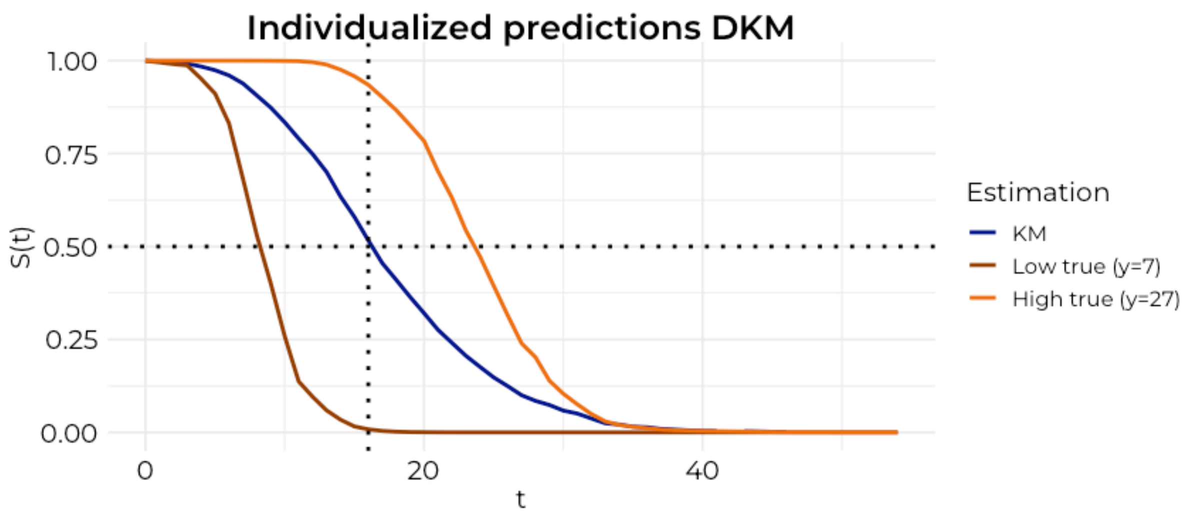
$$\hat{z}_j = \begin{cases} \hat{q}(t_1) & \text{if } j = 1 \\ \hat{z}_{j-1}\hat{q}(t_j) & \text{if } j > 1 \end{cases}$$

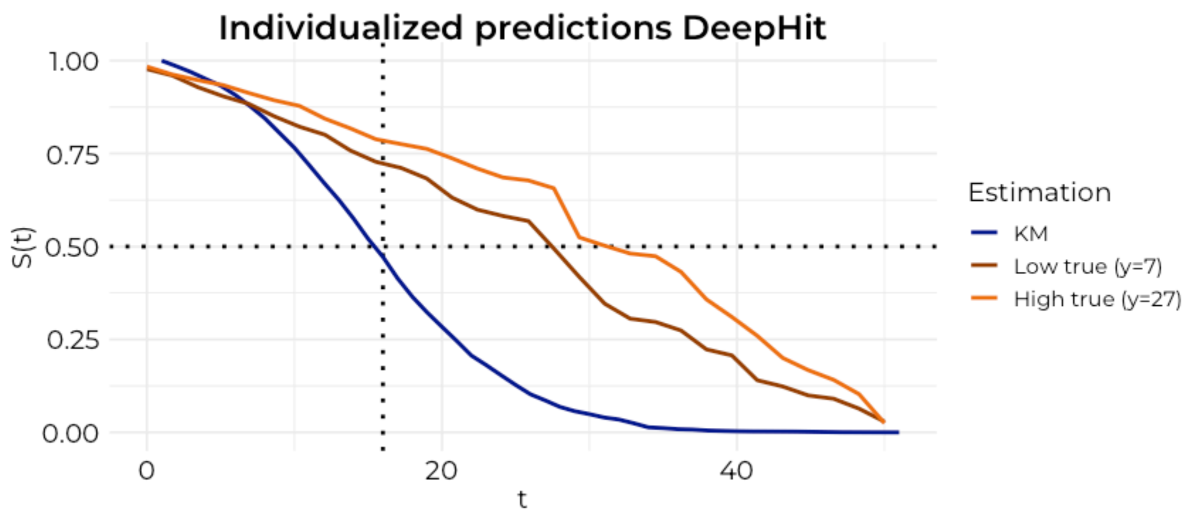


Deep Kaplan-Meier

Individual Predictions

- This allows to construct conditional predictions, without assumptions such as proportional hazards
- Here: a simple example where $\tau_i = \mathcal{G}(x_i^\top \beta, 1)$ and censoring is random

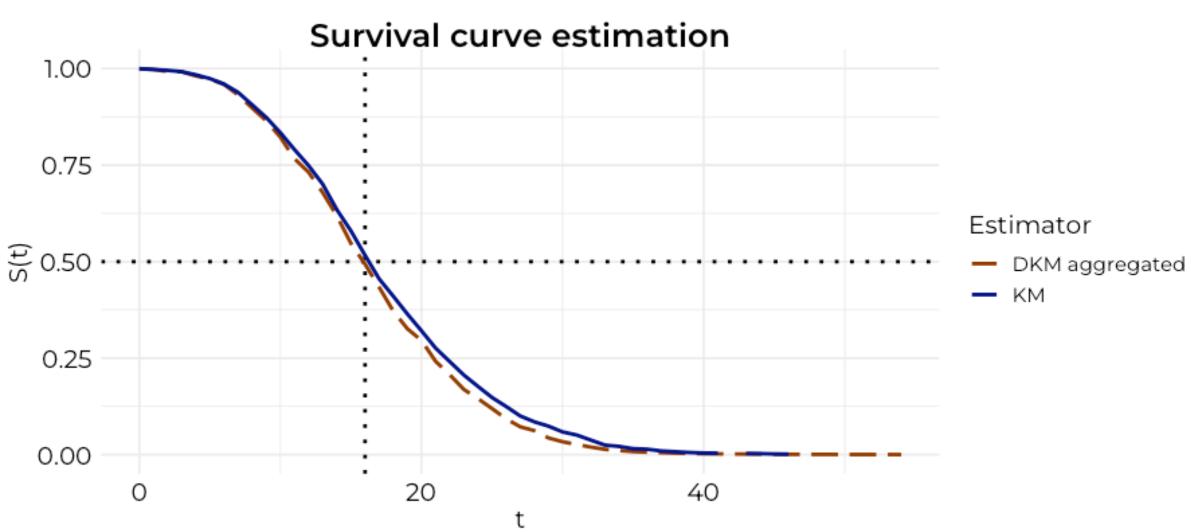


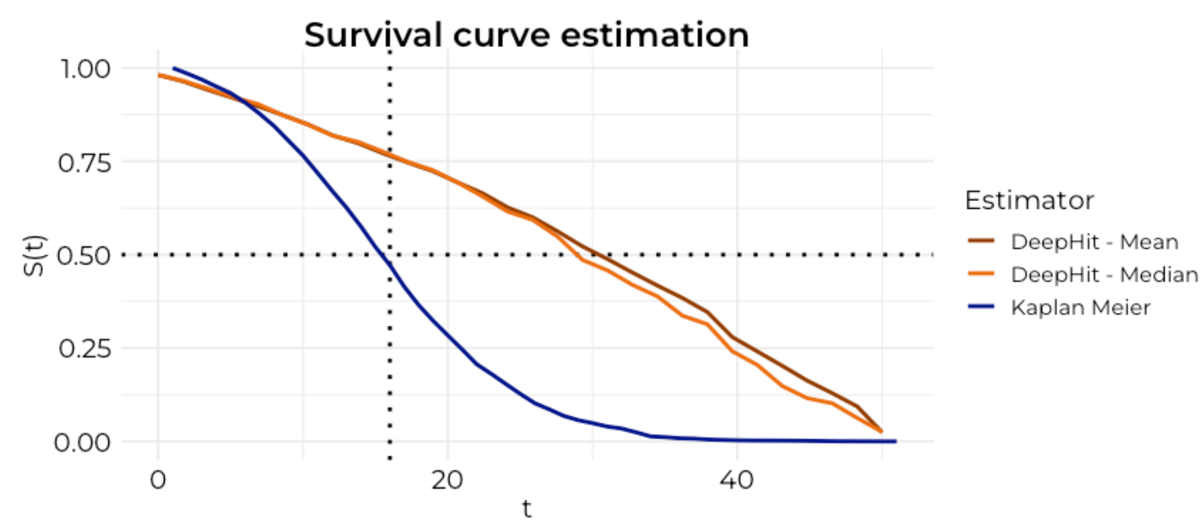


Deep Kaplan-Meier

Averaged Predictions

- But what about (average) calibration? $\mathbb{E}[Y \mid \hat{m}(\boldsymbol{X})] = \hat{m}(\boldsymbol{X})$
- Here: The average prediction





Deep Kaplan-Meier

Random Censoring

- Optimisation is straightforward, unlike in the Cox-Family
- Further, we can show that in expectation, the estimation converges to the Kaplan-Meier estimation

Proposition: The expected value of the DKM is the KM

For a learner that possesses the universal approximation property, the Deep Kaplan-Meier estimator, recovers in expectation the Kaplan-Meier estimation

That is, we show: $\mathbb{E}(\hat{z}_j(x)) = \mathbb{E}\left[\frac{\sum_{k=1}^{j} d_k}{\sum_{k=1}^{j} d_k + r_j - d_j}\right]$

Deep Kaplan-Meier

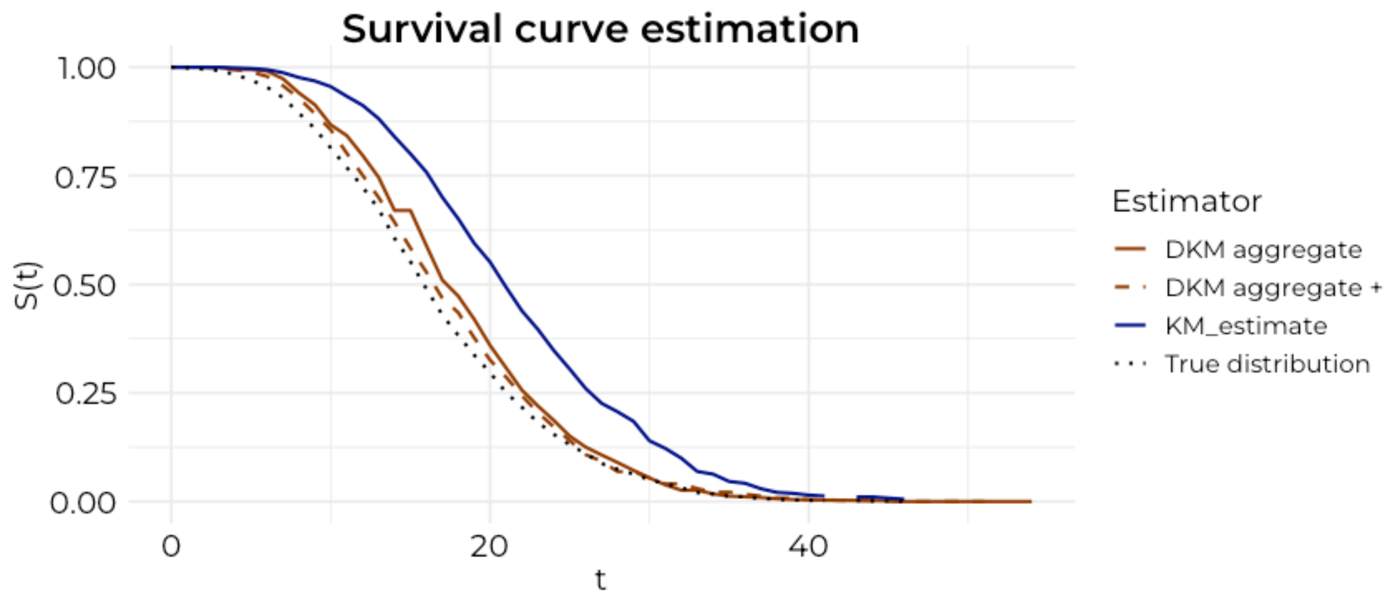
Dependent Censoring

- Random censoring is usually unrealistic
- Consider the case where we have a positive dependence, that is $\mathbb{P}[c_i] = 1 - \min\left\{2 \times \left(\frac{x_i^\top \beta}{\max(x_i^\top \beta)}\right), 1\right\}$

Deep Kaplan-Meier

Dependent Censoring

- Random censoring is usually unrealistic
- Consider the case where we have a positive dependence, that is $\mathbb{P}[c_i] = 1 - \min\left\{2 \times \left(\frac{x_i^\top \beta}{\max(x_i^\top \beta)}\right), 1\right\}$
- Then $\mathbb{E}[z_j] = \mathbb{E}[\hat{z}_j] + \frac{1}{\mathbb{E}[w]}\text{Cov}(w, y)$



Conditioned Kaplan-Meier

Dependent Censoring

- If data that explains the censoring process is available we can include this in the model
- If we are willing to accept conditional independence, we can show that the estimator converges to the true hazard rate

Proposition: Conditional Convergence

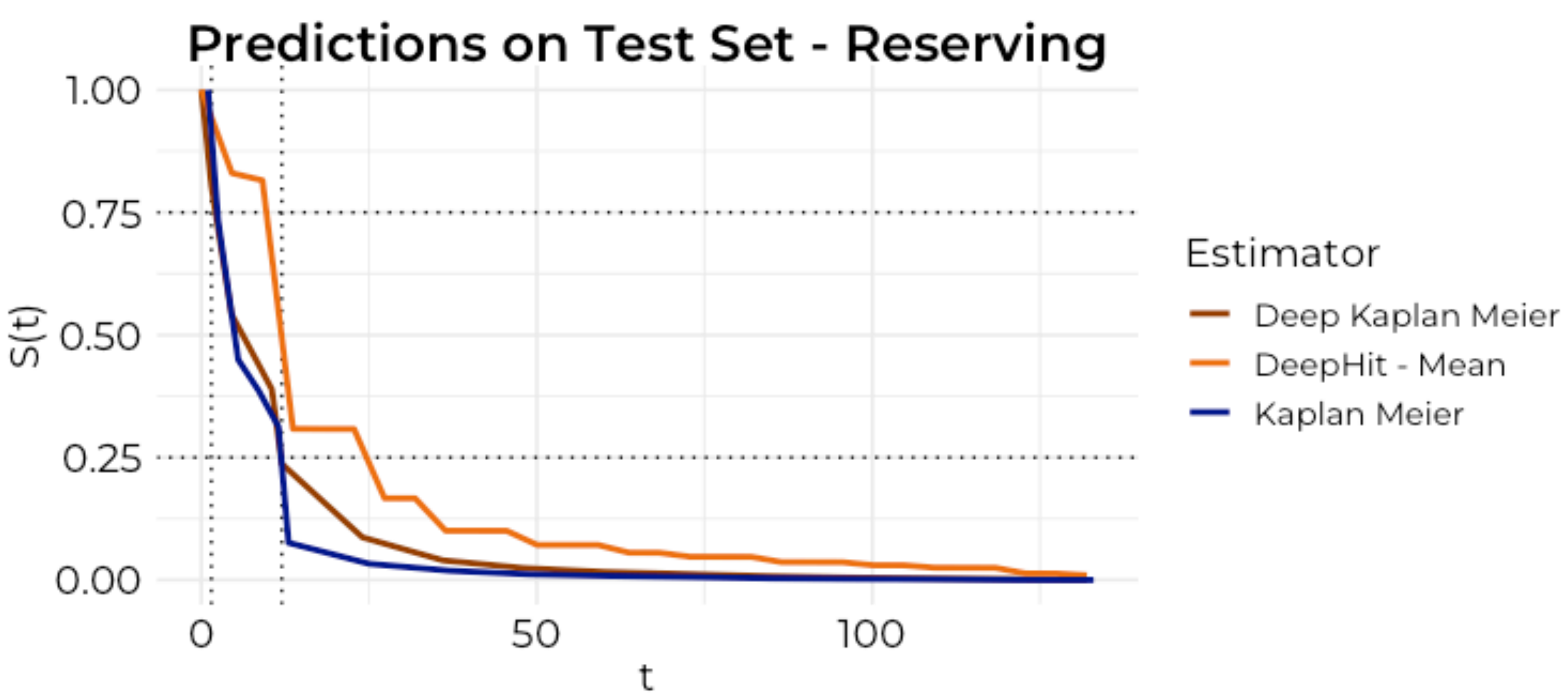
If $h_C(t \mid X, Z, \tau, \tau > t) = h_C(t \mid X, Z, \tau > t)$ holds, then the DKM converges to the true hazard rate, even if censoring and event time are dependent.

Note: Similar to Beran (1981) or more recently example is Chen (2021), this allows to *condition* the Kaplan-Meier estimator, though these approaches use local estimation based on kernels

A (possible) application

In Actuarial Sciences

- As an application - consider Microlevel Reserving - We want to estimate how long a claim will be open (given some covariates)
- Here simulated data from Gabrielli et al. (2018)



In Summary

- Imposing structure into Neural Networks allows to:
  - Recover a flexible version of existing models
  - Have calibrated outputs
  - Safeguards on the estimation allow usage even on small datasets
- Nonlinearities and *conditional* independence enable more realistic estimations
- Many extensions possible, on Quantiles, with included censoring model, etc..

References

Antonio, K. and Plat, R. (2014). Micro-level stochastic loss reserving for general insurance. *Scandinavian Actuarial Journal*, 2014(7):649–669.

Beran, R. (1981). Nonparametric regression with randomly censored survival data.

Chen, G. (2019). Nearest neighbor and kernel survival analysis: Nonasymptotic error bounds and strong consistency rates. In *International Conference on Machine Learning*, pages 1001–1010. PMLR

Craig, E., Zhong, C., and Tibshirani, R. (2021). Survival stacking: casting survival analysis as a classification problem. arXiv preprint arXiv:2107.13480

Gabrielli, A. and V. Wüthrich, M. (2018). An individual claims history simulation machine. *Risks* , 6(2):29.

Kvamme, H. and Borgan, Ø. (2021). Continuous and discrete-time survival prediction with neural networks. *Lifetime data analysis*, 27:710–736

Lee, C., Zame, W., Yoon, J., and Van Der Schaar, M. (2018). Deephit: A deep learning approach to survival analysis with competing risks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32

Willems, S., Schat, A., van Noorden, M., and Fiocco, M. (2018). Correcting for dependent censoring in routine outcome monitoring data by applying the inverse probability censoring weighted estimator. *Statistical Methods in Medical Research*, 27(2):323–335

Appendix: Economic Application

- Hazard of starting a new job, given unemployment benefits