SURVIVAL ANALYSIS FOR HEALTHCARE DATA

From Cox Models to Deep Learning

MAYO CLINIC PLATFORM ACCELERATE EDUCATION SESSION

Andrew Y. K. Foong, Ph.D.

February 25th, 2026

MAYO CLINIC

Radiation Oncology
AI & Data Analytics
AIDA

About me

AI Scientist & Assist. Prof. · Radiation Oncology

Senior Researcher · Microsoft Research

Ph.D. in Machine Learning · Cambridge University

Research Scientist Intern · Google DeepMind

MORE INFO: andrewfoongyk.github.io

Today's talk

1. Survival analysis
   - *Problem statement*
2. Maximum likelihood learning
   - *The objective function*
3. The Cox model
   - *The proportional hazards assumption*
4. DeepSurv
   - *Bringing deep learning to Cox-PH*
5. Discrete-time models
   - *Beyond proportional hazards*
6. Q&A

Survival Analysis

Problem statement

Problem statement

- Define some event of interest, E.g.:
  - Disease progression
  - Death
- We want to predict *who* will experience the event, and *when*
- PROBLEM: for many patients, *we don't observe the event of interest.*
  - Patients drop out ("lost to follow-up")
  - The study ends, and we don't know what will happen to patients in the future
- This called CENSORING, and is ubiquitous in healthcare datasets
- Analyzing this kind of data is the central problem of *survival analysis*

Mathematical description

- Let T_i be the time between some fixed event (e.g., treatment) and the event of interest (e.g., cancer progression) for patient i
  - *For convenience, we can define the treatment time as time zero.*
- Let C_i be the time at which we *stop* following the patient
  - *This is the* CENSORING *time*
- If $C_i > T_i$, then we observe the patient long enough to witness the event.
- If $C_i < T_i$, then we don't observe the event—all we know is that the event has not yet occurred by time C_i.
- We denote whether or not we observe the event by another variable δ_i
  - $\delta_i = 1$ if we observe the event: $C_i > T_i$
  - $\delta_i = 0$ if we don't observe the event: $C_i < T_i$

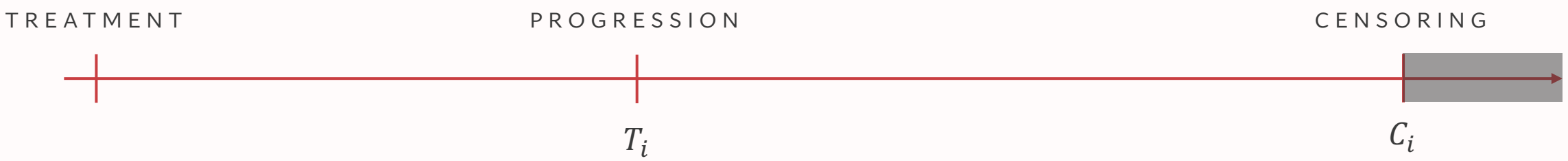
Mathematical description

EXAMPLE 1: *Uncensored data*

EXAMPLE 2: *Censored data*

Mathematical description

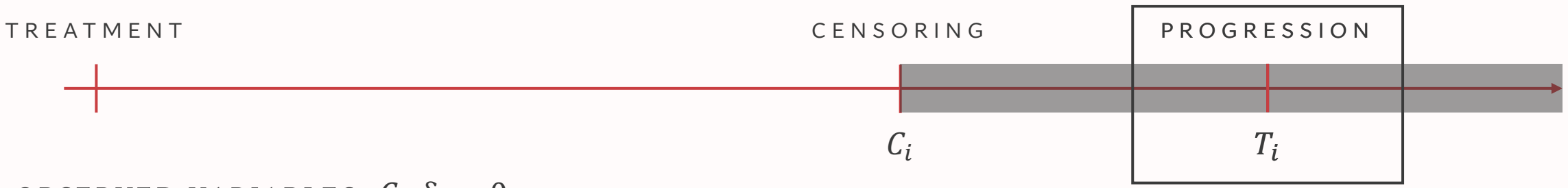
EXAMPLE 1: *Uncensored data*



OBSERVED VARIABLES: $T_i, \delta_i = 1$

EXAMPLE 2: *Censored data*

In a real dataset, we wouldn't see this event!



OBSERVED VARIABLES: $C_i, \delta_i = 0$

Mathematical description

- IDEAL WORLD:
  - Event time T_i
  - Censoring time C_i
- OBSERVATIONAL WORLD:
  - If the event happens first, we observe T_i
  - If censoring happens first, we observe C_i
  - We can summarize this by defining $Y_i = \min(T_i, C_i)$
  - We also observe whether the event occurred first or censoring occurred first, δ_i
  - SUMMARY: in the survival data, we *only* observe (Y_i, δ_i)
- KEY CHALLENGE OF SURVIVAL ANALYSIS:
  - *How do we draw conclusions about T_i, when all we observe is (Y_i, δ_i)?*

Maximum Likelihood Learning

The objective function

The likelihood function

- Can view survival analysis through the lens of the LIKELIHOOD FUNCTION
  - This applies for "classical" methods (Cox-PH) and modern methods (deep learning)
- The likelihood is the *probability of the data given parameters of your model*: $p(\text{data}|\text{parameters})$
  - In our case, the data is $\{(Y_i, \delta_i)\}_{i=1}^N$
  - The *parameters* depend on what model we choose to model the data. We denote them by θ
  - We will also usually have *covariates* x_i for each patient—such as sex, smoking history, risk factors. In that case the likelihood is $p(\text{data}|\text{parameters, covariates})$
- In our notation, this is $p\left(\{(Y_i, \delta_i)\}_{i=1}^N \middle| \{(x_i)\}_{i=1}^N, \theta\right)$
- If each patient is *independent*, this becomes $\prod_{i=1}^N p(Y_i, \delta_i | x_i, \theta)$

Modeling choices

- Two immediate questions:
  1. How do we choose the statistical model? This will determine the form of the likelihood.
  2. For any given model, how do we choose the parameters, θ?
- MODEL SELECTION is probably the most difficult part. It depends on:
  - What assumptions you're willing to make about the data
  - Computational resources, familiarity
- In this talk we will discuss three models:
  - Cox proportional hazards
  - DeepSurv
  - Discrete-time deep learning models
- Thankfully, we can use the same method of estimating θ for each: MAXIMUM LIKELIHOOD

Maximum likelihood learning

- Choose the value of θ that maximizes $p\left(\{(Y_i, \delta_i)\}_{i=1}^N \middle| \{(x_i)\}_{i=1}^N, \theta\right)$
- This can be written:
  - $\prod_{i=1}^N \underbrace{f(y_i|x_i,\theta)^{\delta_i}}_{\text{Probability density of event at time } y_i} \times \underbrace{S(y_i|x_i,\theta)^{1-\delta_i}}_{\text{Probability of not having event before time } y_i}$
  - $S(y_i|x_i,\theta)$ is the *survival function*, defined as:
    - $S(y_i|x_i,\theta) := \Pr(Y_i > y_i|x_i,\theta)$
    - The probability of *not* having experienced the event of interest by time y_i

The hazard function

- It turns out more convenient to write this in terms of the *hazard function*:
  - $h(t|x) = \lim_{\Delta t \to 0} \frac{\Pr(t+\Delta t > T > t \mid T \geq t, x)}{\Delta t}$
  - INTERPRETATION: the hazard function is the probability density of having the event occur *at* time t, *conditional* on *not* having had the event *before* time t

THEOREM 1: *The hazard function and the survival function are related in the following way*:

$$S(t|x) = \exp(-\int_0^t h(u|x)\,\mathrm{d}u).$$

PROOF SKETCH: write the survival function as a product of conditional survivals over small intervals Δt, then take the limit $\Delta t \to 0$.

Likelihood to hazard

- Theorem 1 implies that we can write the likelihood entirely in terms of the hazard function: *once we specify the hazard, we specify the entire model.*
- After some algebra and rearranging, we get that the likelihood equals:
  - $\prod_{i=1}^{N} h(y_i|x_i,\theta)^{\delta_i} \times \exp(-\int_0^{y_i} h(u|x,\theta)\,\mathrm{d}u)$
- It's usually easier to work in terms of the log-likelihood:
  - $\sum_{i=1}^{N} \delta_i \log h(y_i|x_i,\theta) - \int_0^{y_i} h(u|x,\theta)\,\mathrm{d}u$
- INTERPRETATION:
  - The second term is the probability of surviving until y_i
  - If the patient is censored, the first term is zero—no contribution
  - If not, the first term is $h(y_i|x_i,\theta)$: the probability of having the event at y_i given they survived to that point.

The Cox Model

The proportional hazards assumption

The proportional hazards assumption

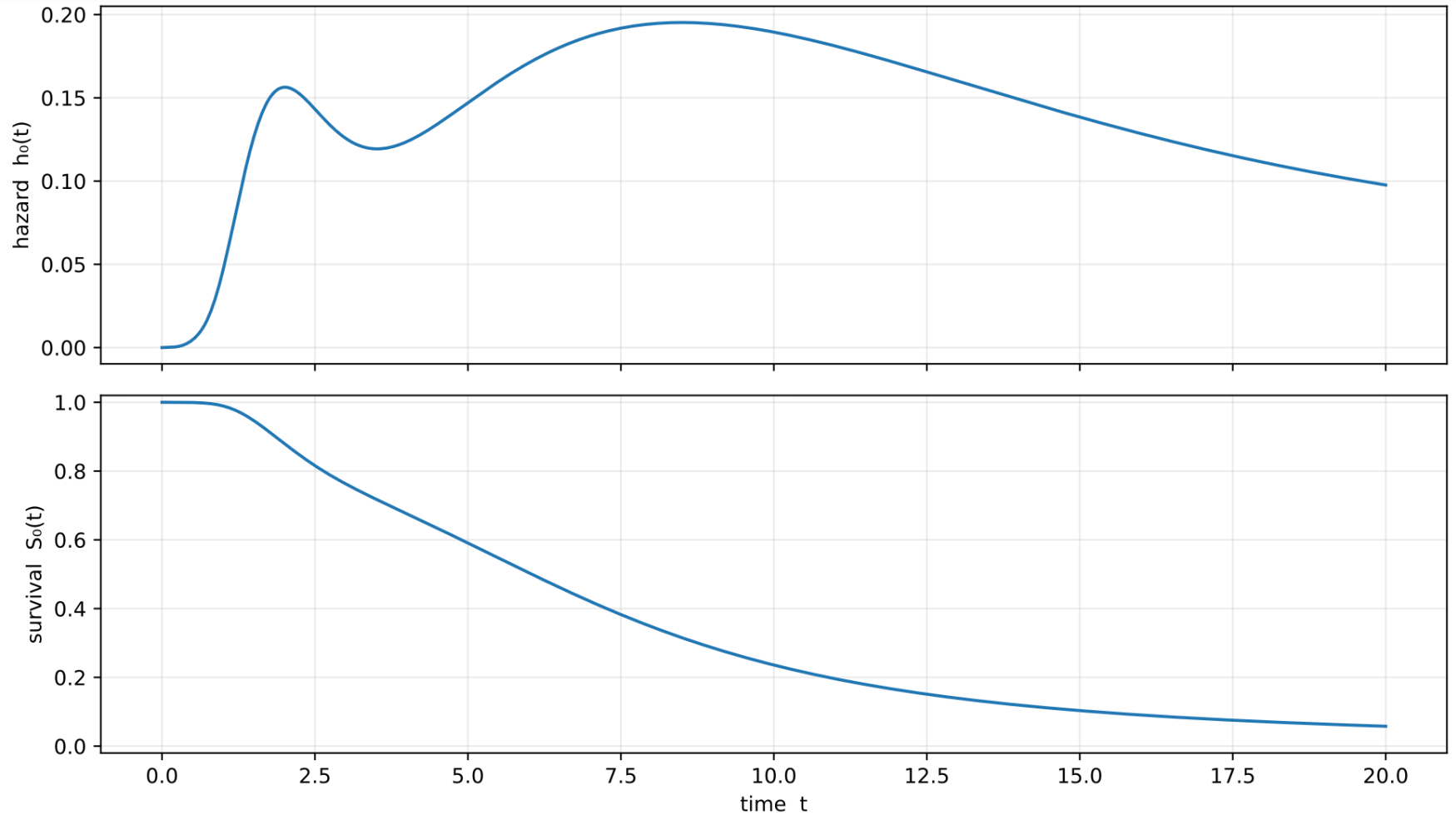
- Up to this point, we have not made *any* assumptions on the form of the model—this applies to classical Cox as well as deep learning methods.
- We now make the *proportional hazards assumption*:

COX PROPORTIONAL HAZARDS (COX-PH): *Assume that the hazard function can be written as*:

$$h(t|x,\theta) = h_0(t)\exp(\theta^{\mathrm{T}}x),$$

where $h_0(t)$ *is any arbitrary function of time that does* not *depend on* x*, known as the* BASELINE HAZARD. *The parameters* θ *are more commonly denoted* β *and are known as* HAZARD RATIOS.

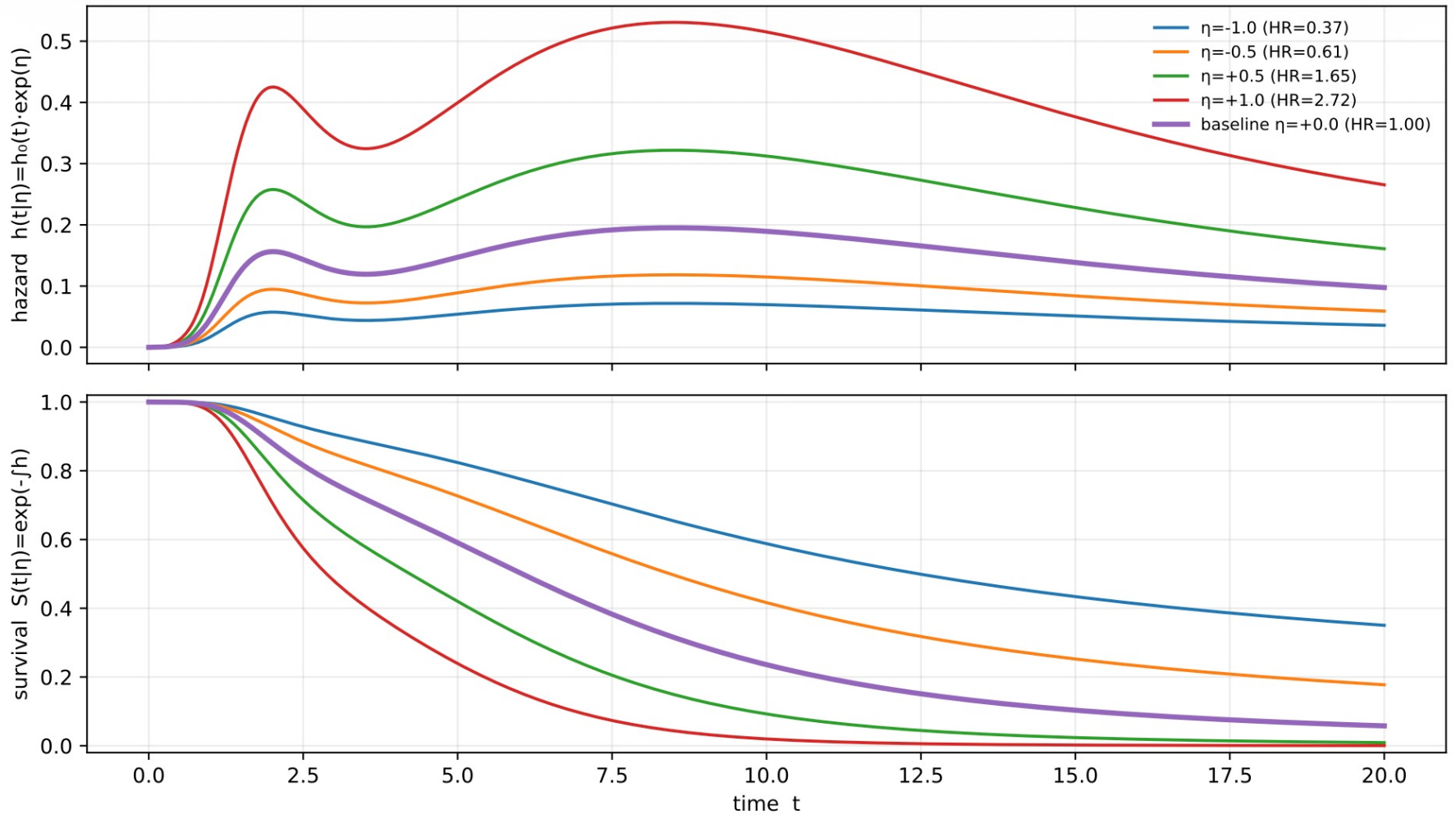
Understanding proportional hazards



Some arbitrary, randomly generated baseline hazard function $h_0(t)$

The corresponding baseline survival function calculated using Theorem 1, $\exp(-\int_0^t h_0(u)\,\mathrm{d}u)$

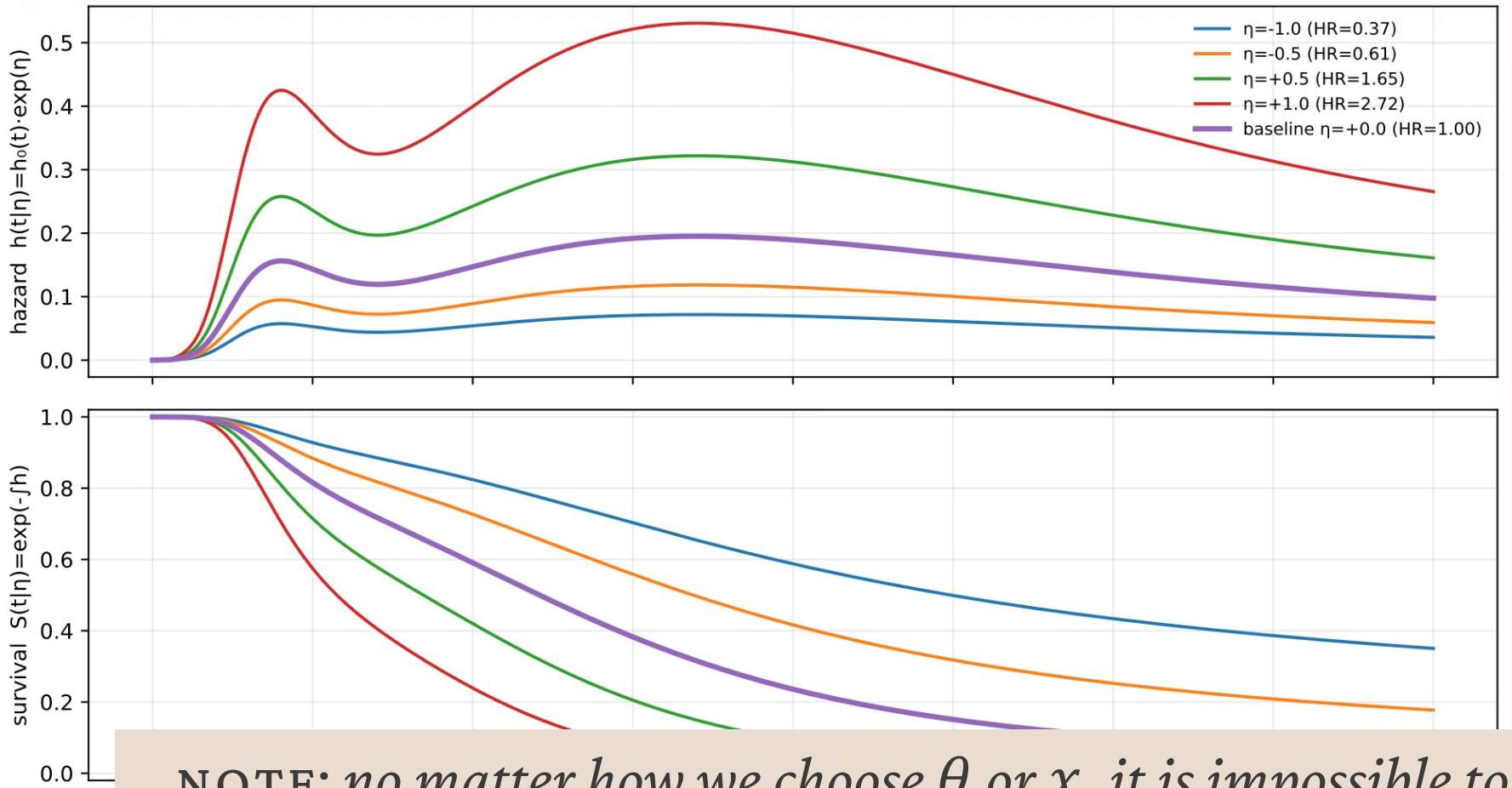
Understanding proportional hazards



Hazard functions generated by varying the value of $\eta = \exp(\theta^{\mathrm{T}} x)$ in $h_0(t)\exp(\theta^{\mathrm{T}} x)$

The corresponding survival functions calculated using Theorem 1, $\exp(-\int_0^t h(u|x,\theta)\,\mathrm{d}u)$

Understanding proportional hazards



Hazard functions generated by varying the value of $\eta = \exp(\theta^{\mathrm{T}} x)$ in $h_0(t)\exp(\theta^{\mathrm{T}} x)$

The corresponding survival functions calculated using Theorem 1, $\exp(-\int_0^t h(u|x,\theta)\,\mathrm{d}u)$

NOTE: *no matter how we choose θ or x, it is impossible to make survival curves cross each other under the proportional hazards assumption.*

The Cox likelihood

- There's still one problem: *how do we learn the value of the baseline hazard, $h_0(t)$, and the hazard ratios, θ?*
- Recall the *maximum likelihood objective*:
  - Want to find θ to optimize $L(\theta) = \sum_{i=1}^{N} \delta_i \log h(y_i|x_i, \theta) - \int_0^{y_i} h(u|x, \theta)\, \mathrm{d}u$
  - Substitute in $h(t|x, \theta) = h_0(t) \exp(\theta^{\mathrm{T}} x)$ into $L(\theta)$
  - After some algebra (skipped here), we find that the likelihood only depends on the baseline hazard at a *finite* number of timepoints: the observed event times
  - Maximizing with respect to the values of $h_0(t)$ at those times, and substituting those back into the likelihood, we can *eliminate $h_0(t)$ entirely*

KEY TAKEAWAY: the Cox-PH model requires learning the hazard ratios *and* the baseline hazard function. Fortunately, we can *algebraically* optimize the likelihood over *all possible* baseline hazards functions—so in practice we only need to learn the hazard ratios.

The Cox likelihood

- Substituting the optimal values of the baseline hazard function into the likelihood, we obtain the *Cox partial likelihood*:
  - $\prod_{j=1}^{D} \frac{\exp(\theta^{\mathrm{T}} x_{i(j)})}{\sum_{k \in R_j} \exp(\theta^{\mathrm{T}} x_k)}$
  - D is the number of *observed* events
  - $i(j)$ is the index of the patient that has the jth observed event
  - R_j is the jth *risk set*: i.e., the set of all patients who have *not* been censored *or* had the event by the time of the jth event
- INTERPRETATION:
  - R_j is the set of all people who *could* have had the jth event
  - $\exp(\theta^{\mathrm{T}} x_{i(j)})$ is the relative hazard of the person who *actually* had that event
  - The Cox partial likelihood is the product of all the normalized relative hazards for the actually observed events

The Cox likelihood

- Given the Cox partial likelihood (which is easy to compute):
  - $\prod_{j=1}^{D} \frac{\exp(\theta^{\mathrm{T}} x_{i(j)})}{\sum_{k \in R_j} \exp(\theta^{\mathrm{T}} x_k)}$
- We now need to optimize it with respect to θ
- This is a convex optimization problem, easily solved using the Newton–Raphson method
- Implementations widely available in R and Python libraries

DeepSurv

Bringing deep learning to Cox-PH

DeepSurv

- DeepSurv (2018) is a Cox-PH model where the linear predictor $\theta^{\mathrm{T}} x$ is replaced by a deep neural network:

DEEPSURV: *Assume that the hazard function can be written as*:

$$h(t|x,\theta) = h_0(t)\exp(f_\theta(x)),$$

where $h_0(t)$ *is any arbitrary function of time that does* not *depend on* x*, known as the* BASELINE HAZARD*, and* f_θ *is a deep neural network with weights* θ*.*

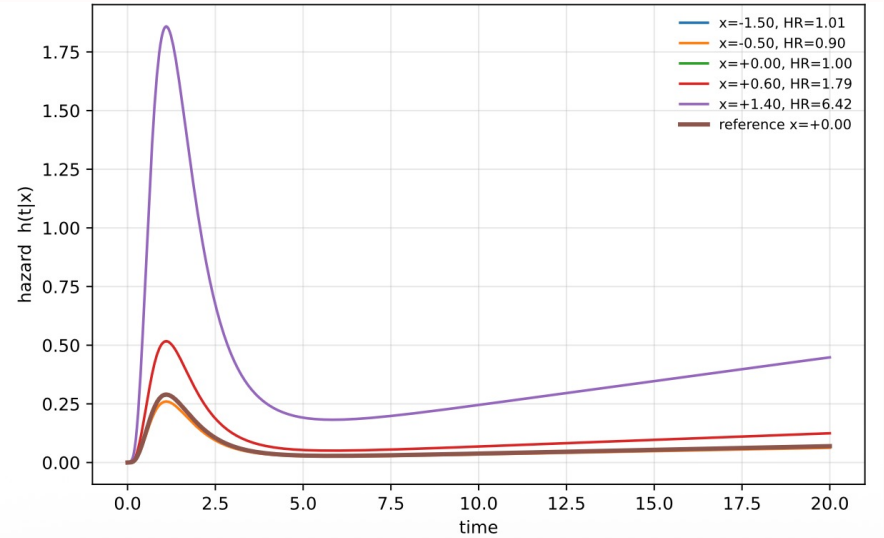
Katzman, Jared L., Uri Shaham, Alexander Cloninger, Jonathan Bates, Tingting Jiang, and Yuval Kluger. "DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network." *BMC medical research methodology* 18, no. 1 (2018): 24.

DeepSurv

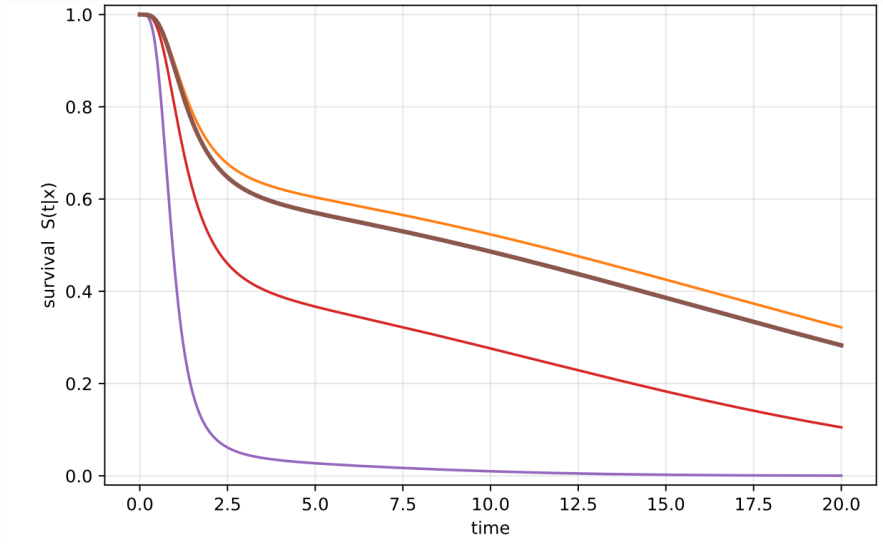
- ADVANTAGE: since we maintain the proportional hazards assumption, we can use *the same form* of objective function as in Cox-PH:
  - $\prod_{j=1}^{D} \frac{\exp(f_\theta(x_{i(j)}))}{\sum_{k \in R_j} \exp(f_\theta(x_k))}$
- DISADVANTAGE: still maintains the proportional hazards assumption. (Which is why we might have wanted a more flexible model in the first place.)

Katzman, Jared L., Uri Shaham, Alexander Cloninger, Jonathan Bates, Tingting Jiang, and Yuval Kluger. “DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network.” *BMC medical research methodology* 18, no. 1 (2018): 24.

Non-linear proportional hazards



Hazard functions generated by varying the value of $f_\theta(x)$ in $h_0(t)\exp(f_\theta(x))$



The corresponding survival functions calculated using Theorem 1, $\exp(-\int_0^t h(u|x,\theta)\,\mathrm{d}u)$

Discrete-time Models

Beyond proportional hazards

Discrete-time models

- To go beyond the proportional hazards assumption, the easiest method is to *discretize time into bins.*

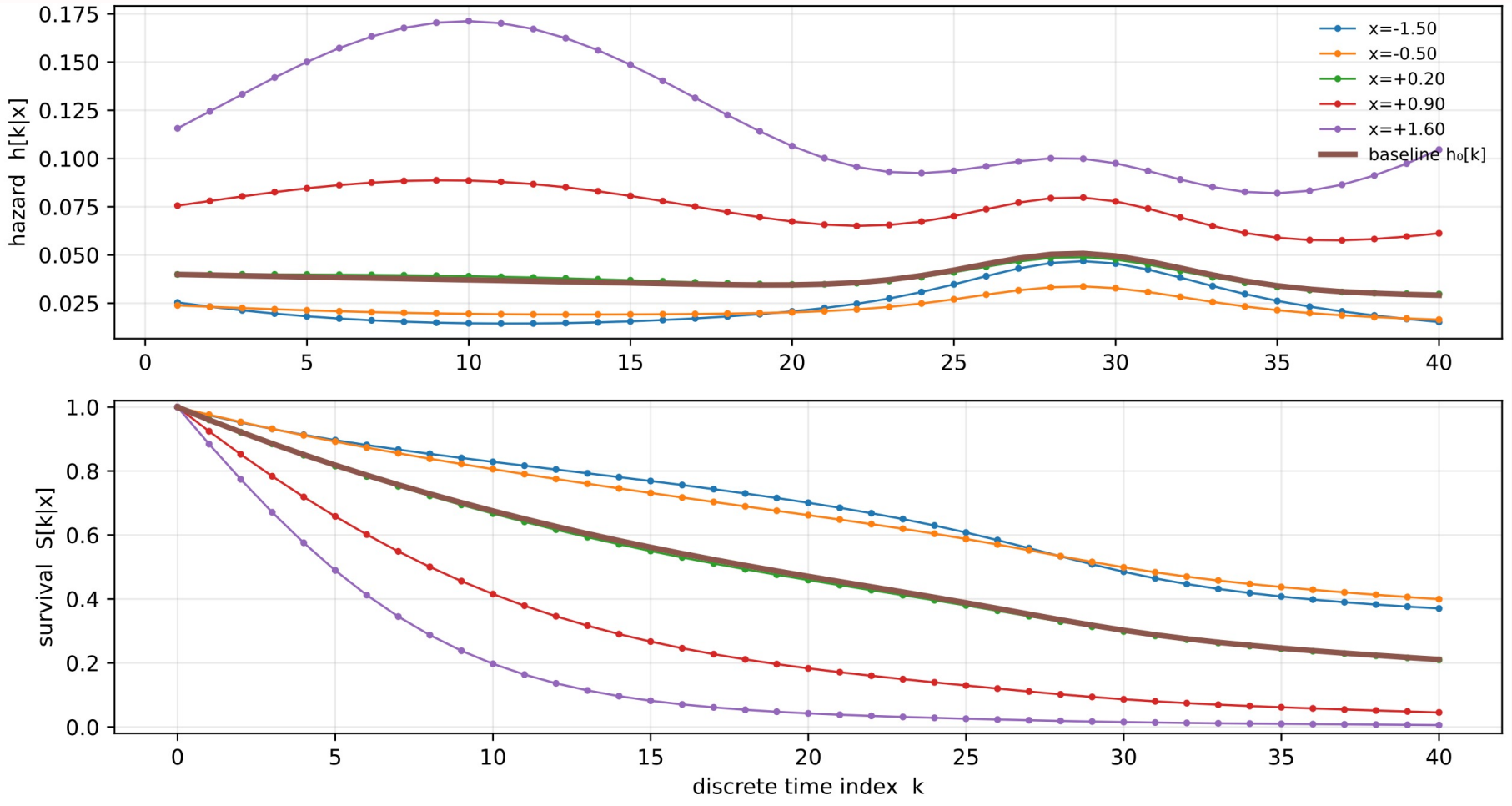
- Define the *discrete-time hazard function* as:
  - $h_{ik} = \Pr(T \text{ in bin } k \mid T \text{ not in bins 1 to } k-1, x_i)$
- Analogously to the continuous case, the likelihood function can be written entirely in terms of h_{ik}:
  - $L_i = \left[\prod_{k=1}^{t_i-1}(1-h_{ik})\right](h_{it_i})^{\delta_i}(1-h_{it_i})^{1-\delta_i}$
  - The full likelihood is the product of L_i over all patients

Discrete-time models

- We can easily compute the likelihood $L_i = \left[\prod_{k=1}^{t_i-1}(1-h_{ik})\right](h_{it_i})^{\delta_i}(1-h_{it_i})^{1-\delta_i}$ *provided* we specify h_{ik}.
- Idea: parameterize h_{ik} with a deep neural network:
  - $h_{ik} = f_\theta(x_i, k)$
  - Learn θ by maximum likelihood
- This is the idea behind Nnet-survival (2018)

Gensheimer, Michael F., and Balasubramanian Narasimhan. “A scalable discrete-time survival model for neural networks.” *PeerJ* 7 (2019): e6257.

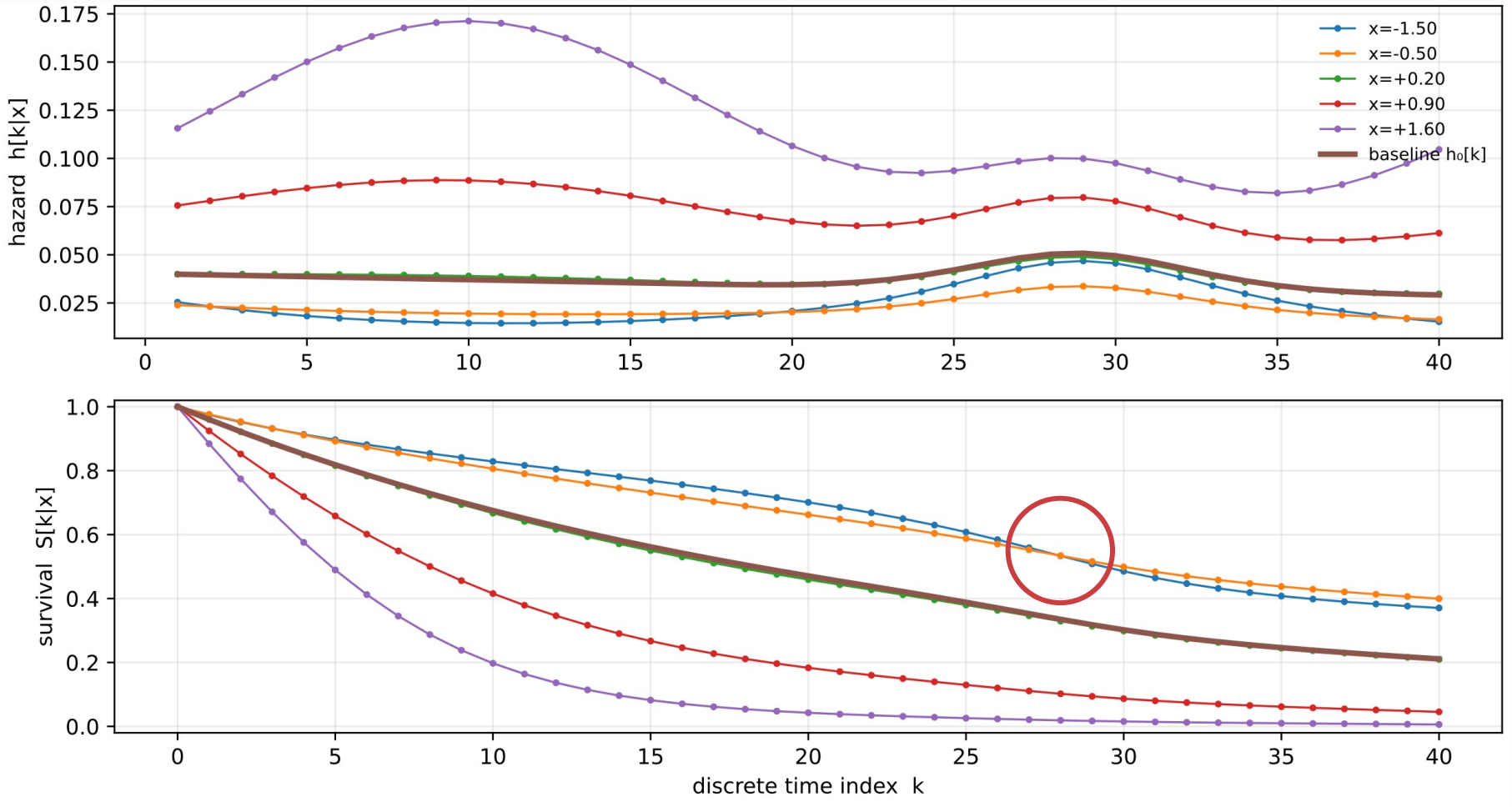
Non-proportional discrete-time hazards



Discrete-time hazard functions generated by varying the inputs to a neural network

The corresponding survival functions calculated using a discrete—time analogue of Theorem 1

Non-proportional discrete-time hazards



Discrete-time hazard functions generated by varying the inputs to a neural network

The corresponding survival functions calculated using a discrete—time analogue of Theorem 1

Question & Answer