BoXHED: Boosted eXact Hazard Estimator with Dynamic covariates

Xiaochen Wang
Yale University

With Donald K.K. Lee (Emory U.), Bobak J. Mortazavi (TAMU),
Arash Pakbin (TAMU), Hongyu Zhao (Yale U.)

Motivation

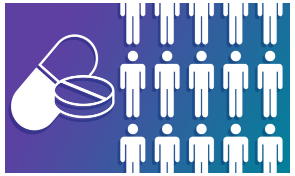
Dynamic Features
in Survival Analyses

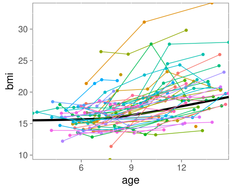
Motivation

High frequency health vitals in ICU

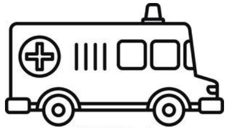
Dynamic Features in Survival Analyses

Motivation





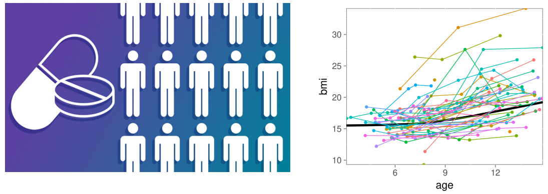
Longitudinal data from clinical studies



High frequency health vitals in ICU

Dynamic Features in Survival Analyses

Motivation

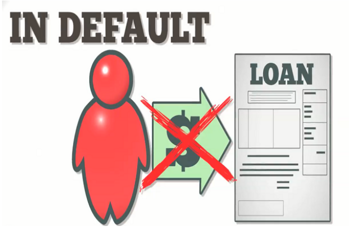


Longitudinal data from
clinical studies

High frequency health
vitals in ICU

Dynamic Features in Survival Analyses

Mobile data and
wearables devices



Behavioral data in
financial risk assessment

Challenges & Our contributions

- Challenges:
  - ML survival methods mainly focus on time-static features. (Ishwaran et al. 08; Ranganath et al. 16; Bellot & van der Schaar 18, 19; Lee et al. 19)
  - Methods dealing with dynamic features are very sparse:
    - **Non-parametric:** kernel smoothing for low-dimensional covariate settings.
    - **Parametric**: ‘flexsurv’ R package.

Challenges & Our contributions

- Challenges:
  - ML survival methods mainly focus on time-static features. (Ishwaran et al. 08; Ranganath et al. 16; Bellot & van der Schaar 18, 19; Lee et al. 19)
  - Methods dealing with dynamic features are very sparse:
    - **Non-parametric:** kernel smoothing for low-dimensional covariate settings.
    - **Parametric**: ‘flexsurv’ R package.
- Contributions:
  1. First publicly available software for boosted hazard estimation with time-dependent features.

     https://github.com/BoXHED
  2. Novel algorithmic implementation of **Lee, Chen, Ishwaran “*Boosted nonparametric hazards with time-dependent covariates*” (2017)**

Problem statement

Each participant i is represented by a triplet $(X_i(t)_{t\in[0,T_i]}, \Delta_i, T_i)$.

- $X_i(t)$ is a set of continuously-monitored features.
- Δ_i is a binary event indicator: 1 for an uncensored instance and 0 for a censored instance.
- T_i is the observed time, i.e.

$$T_i = \begin{cases} \text{Event time} & \text{if } \Delta_i = 1 \\ \text{Censoring time} & \text{if } \Delta_i = 0 \end{cases}$$

Goal: Given above information of n participants, we want to estimate log-hazard function $F(t, x)$.

Loss function

- **Loss function** – negative log-likelihood.

$$R(F) = \frac{1}{n}\sum_{i=1}^{n}\left\{\int_0^{T_i} e^{F(t,X_i(t))}dt \; - \Delta_i F(T_i, X_i(T_i))\right\}$$

- **Challenge:** Likelihood risk $R(F)$ is too complex to be optimized using traditional techniques. Solution provided in **Lee, Chen, Ishwaran 17**.

Algorithm Overview

Algorithm: BoXHED

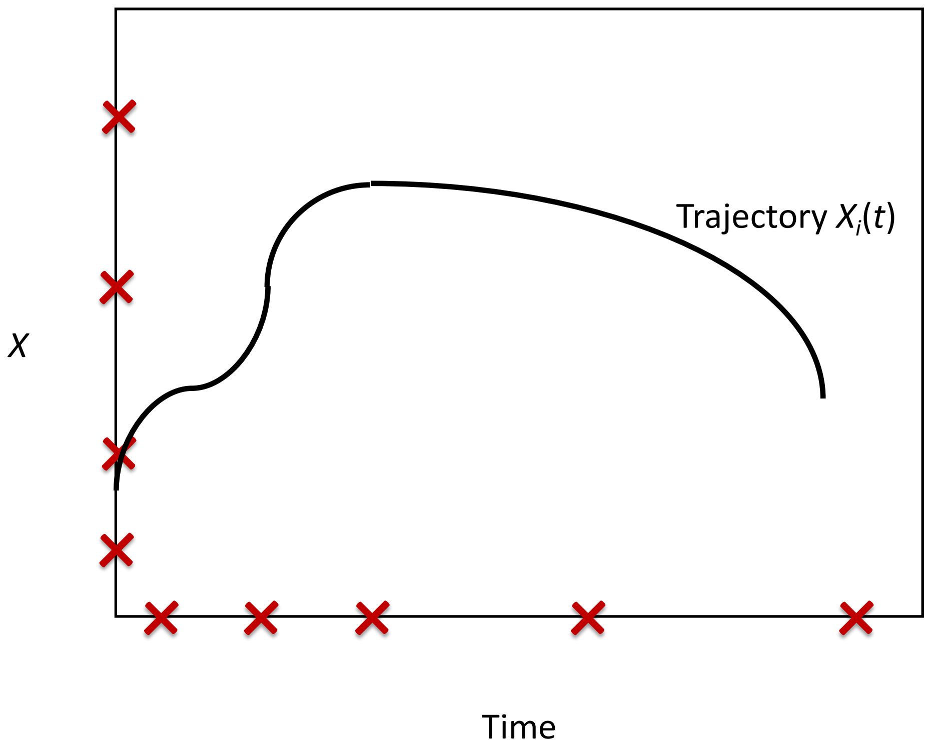
Input: n samples $\{(X_i(t), T_i, \Delta_i)_{i=1}^n | 0 \leq t \leq T_i\}$, maximum # of iterations M, maximum # of splits L, and step size ν.
Initialize $\hat{F}_0 = \log(\frac{\sum_{i=1}^n \Delta_i}{\sum_{i=1}^n T_i})$.
Propose candidate splits on time and features.
for $m = 1$ **to** M **do**
    Compute the tree g_m that minimizes likelihood risk.
    Update log-hazard $\hat{F}_m \leftarrow \hat{F}_{m-1} - \nu g_{m-1}$.
end for

Algorithm Overview

Algorithm: BoXHED

Input: n samples $\{(X_i(t), T_i, \Delta_i)_{i=1}^n | 0 \leq t \leq T_i\}$, maximum # of iterations M, maximum # of splits L, and step size ν.
Initialize $\hat{F}_0 = \log(\frac{\sum_{i=1}^n \Delta_i}{\sum_{i=1}^n T_i})$.
Propose candidate splits on time and features.
for $m = 1$ **to** M **do**
    Compute the tree g_m that minimizes likelihood risk.
    Update log-hazard $\hat{F}_m \leftarrow \hat{F}_{m-1} - \nu g_{m-1}$.
end for

Constructing the tree g_m

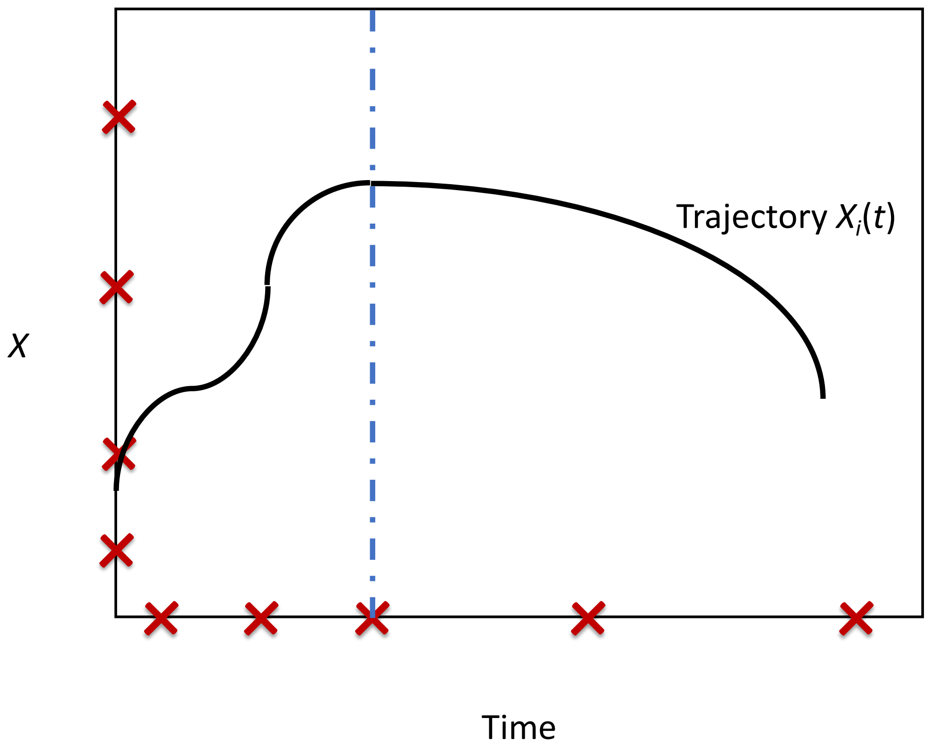


× Candidate splits on time and feature

<u>Tree Construction Demo</u>

- Select candidate splits based on percentiles (adjustable).

Constructing the tree g_m

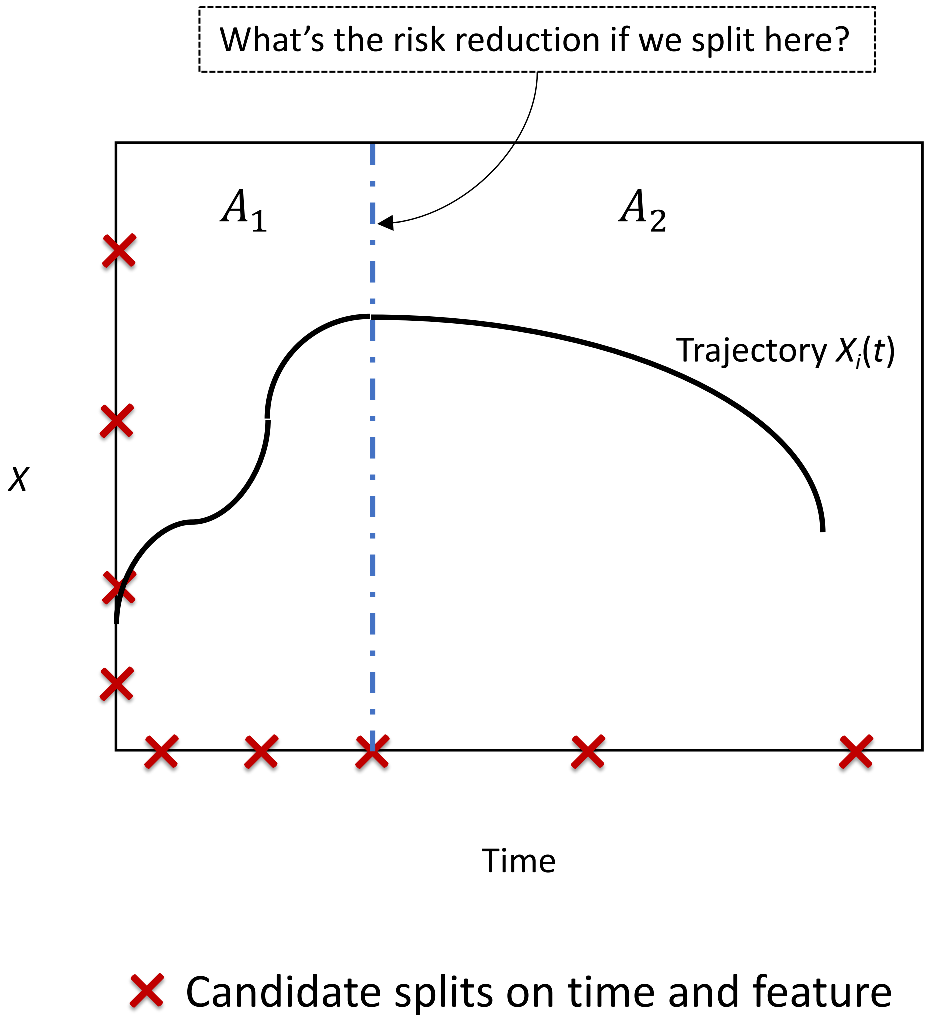


✕ Candidate splits on time and feature

<u>Tree Construction Demo</u>

- Select candidate splits based on percentiles (adjustable).
- Splits chosen to minimize R(F). How?

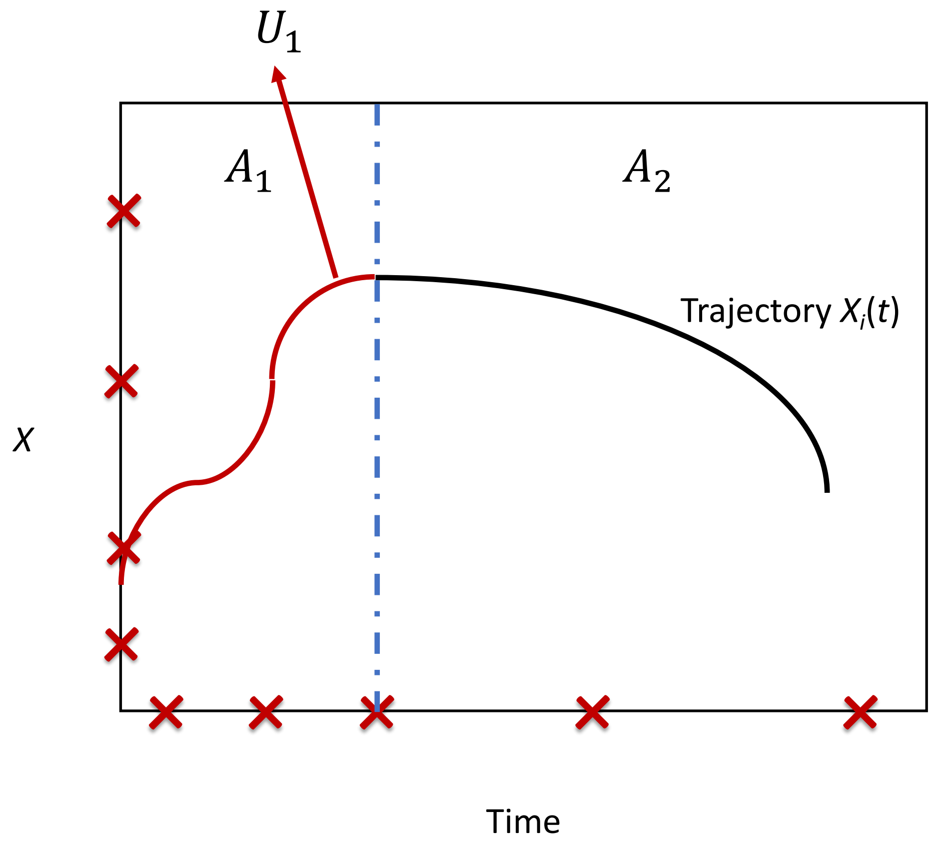
Constructing the tree g_m



<u>Tree Construction Demo</u>

- Select candidate splits based on percentiles (adjustable).
- Splits chosen to minimize R(F). How?
    1. A split creates two new sub-regions A_1 and A_2.

Constructing the tree g_m



Candidate splits on time and feature

Tree Construction Demo

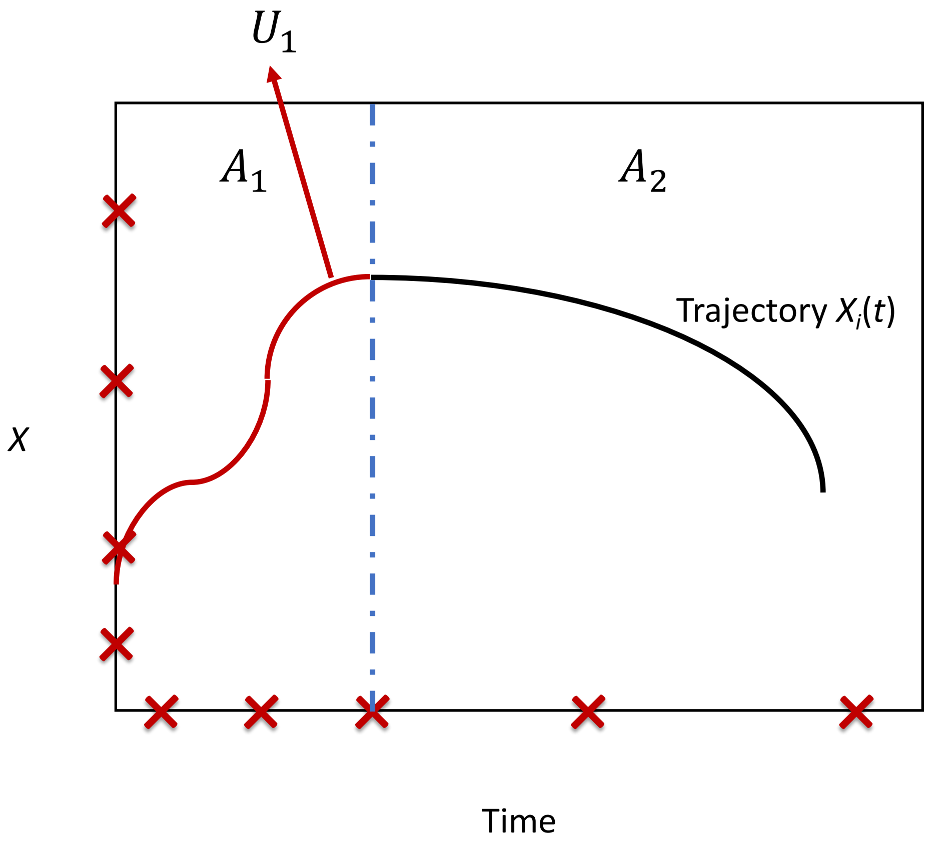
- Select candidate splits based on percentiles (adjustable).
- Splits chosen to minimize R(F). How?
  1. A split creates two new sub-regions A_1 and A_2.
  2. Split score:

$$d = \sum_{k=1}^{2} V_k \left(1 + \log \frac{U_k}{V_k}\right) - (V_1 + V_2)(1 + \log \frac{U_1+U_2}{V_1+V_2}) \text{, where}$$

$$U_k = \sum_{i=1}^{n} \int_0^{T_i} e^{F_m(t, X_i(t))} I_{A_k}\big(t, X_i(t)\big) dt,$$

$$V_k = \#\ of\ observed\ events\ in\ A_k.$$

Constructing the tree g_m



✕ Candidate splits on time and feature

<u>Tree Construction Demo</u>

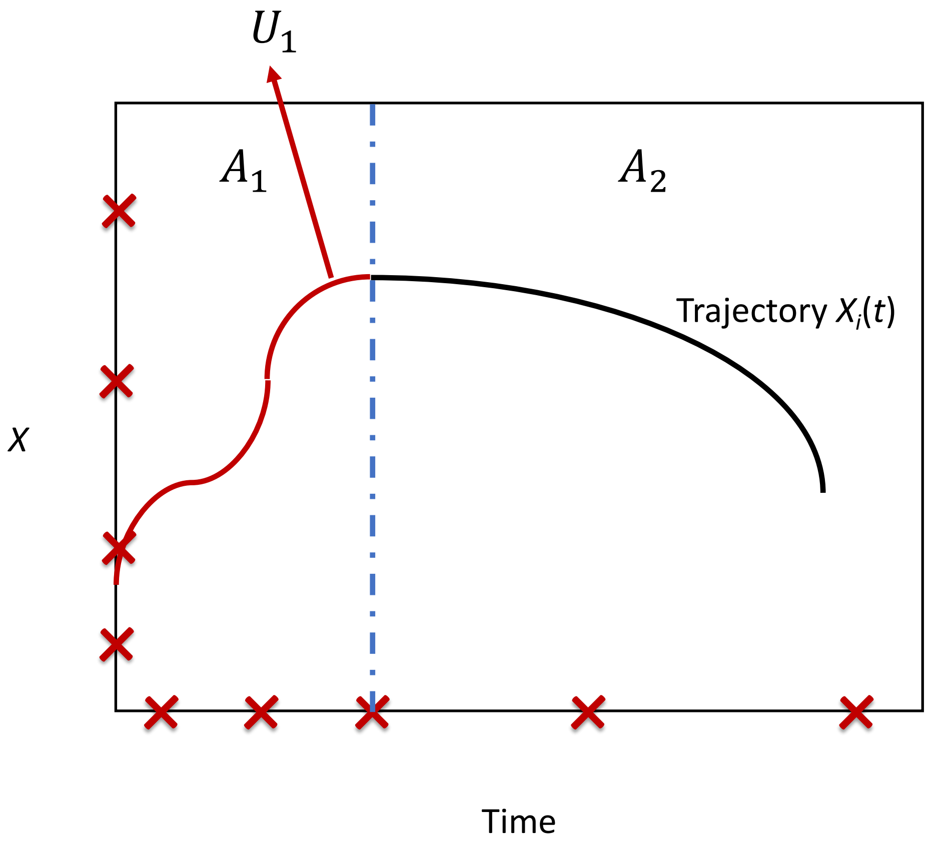
- Select candidate splits based on percentiles (adjustable).
- Splits chosen to minimize R(F). How?
  1. A split creates two new sub-regions A_1 and A_2.
  2. Split score:

$$d = \sum_{k=1}^{2} V_k \left(1 + \log \frac{U_k}{V_k}\right) - (V_1 + V_2)(1 + \log \frac{U_1+U_2}{V_1+V_2}) \text{, where}$$

$$U_k = \sum_{i=1}^{n} \int_0^{T_i} e^{F_m(t, X_i(t))} I_{A_k}(t, X_i(t)) dt,$$
$$V_k = \# \ of \ observed \ events \ in \ A_k.$$

  3. Choose the split that minimized d.

Constructing the tree g_m



✕ Candidate splits on time and feature

Tree Construction Demo

- Select candidate splits based on percentiles (adjustable).
- Splits chosen to minimize R(F). How?
    1. A split creates two new sub-regions A_1 and A_2.
    2. Split score:

$$d = \sum_{k=1}^{2} V_k \left(1 + \log\frac{U_k}{V_k}\right) - (V_1 + V_2)(1 + \log\frac{U_1+U_2}{V_1+V_2})$$, where

$$U_k = \sum_{i=1}^{n} \int_0^{T_i} e^{F_m(t, X_i(t))} I_{A_k}\big(t, X_i(t)\big) dt,$$
$$V_k = \#\ of\ observed\ events\ in\ A_k.$$

    3. Choose the split that minimized d.
- Choose subsequent splits to also minimize split score.

Results

- Simulation data
- Framingham heart study data

Simulation Data

Four hazard functions (Pérez et al. 13)

$$\lambda_1(t, x_t) = Beta(t, 2, 2) \times Beta(x_t, 2, 2), \qquad t \in (0, 1];$$

$$\lambda_2(t, x_t) = Beta(t, 4, 4) \times Beta(x_t, 4, 4), \qquad t \in (0, 1];$$

$$\lambda_3(t, x_t) = \frac{1}{t} \frac{\phi(\log t - x_t)}{\Phi(x_t - \log t)}, \qquad t \in (0, 5];$$

$$\lambda_4(t, x_t) = \frac{3}{2} t^{0.5} \exp\left(-\frac{1}{2}\cos(2\pi x_t) - \frac{3}{2}\right), \qquad t \in (0, 5].$$

0, 20, and 40 irrelevant features from standard normal distribution are added to above four hazards.

Methods

	Can handle time-dependent features?	Nonparametric?	Variable selection	Parameter tuning
BoXHED	√	√	√	Cross-validated on training data
Kernel Smoothing	√	√		Kernel bandwidth tuned directly to *test data*
FlexSurv	√		√	Best parametric family for *test data*
Black-boost			√	Best parametric family and #iterations for *test data*

RMSE error

Hazard	#Irrelevant covariates	Estimator			
		BoXHED	kernel	flexsurv	blackboost
λ_1	0	0.17 (0.17, 0.17)	**0.14 (0.14, 0.15)**	0.53 (0.52, 0.54)	0.58 (0.57, 0.59)
	20	**0.20 (0.20, 0.20)**	3.4 (3.0, 3.9)	0.54 (0.53, 0.54)	0.58 (0.57, 0.59)
	40	**0.21 (0.20, 0.21)**	43 (5.7, 80)	0.54 (0.54, 0.55)	0.58 (0.57, 0.59)
λ_2	0	0.23 (0.23, 0.24)	**0.11 (0.11, 0.12))**	1.1 (1.1, 1.1)	1.4 (1.4, 1.4)
	20	**0.25 (0.25, 0.26)**	4.5 (3.9, 5.2)	1.1 (1.1, 1.1)	1.4 (1.4, 1.4)
	40	**0.26 (0.26, 0.27)**	29 (11, 46)	1.1 (1.1, 1.1)	1.4 (1.4, 1.4)
λ_3	0	0.038 (0.037, 0.040)	0.046 (0.044, 0.049)	**0.0040 (0.0039, 0.0041)**	0.10 (0.10, 0.11)
	20	0.047 (0.046, 0.049)	1.8 (1.1, 2.5)	**0.020 (0.019, 0.020)**	0.10 (0.10, 0.11)
	40	0.050 (0.048, 0.051)	7.6 (5.3, 9.7)	**0.030 (0.029, 0.031)**	0.10 (0.10, 0.11)
λ_4	0	0.049 (0.048, 0.050)	**0.045 (0.044, 0.046)**	0.20 (0.19, 0.20)	0.20 (0.19, 0.20)
	20	**0.060 (0.059, 0.062)**	3.9 (0.66, 7.1)	0.20 (0.19, 0.20)	0.20 (0.19, 0.20)
	40	**0.069 (0.067, 0.070)**	5.5 (4.3, 6.7)	0.20 (0.20, 0.21)	0.20 (0.19, 0.20)

RMSE error with 95% confidence interval.

RMSE error

The kernel function is a beta density, resembling λ_1 and λ_2.

Hazard	#Irrelevant covariates	Estimator: BoXHED	kernel	flexsurv	blackboost
λ_1	0	0.17 (0.17, 0.17)	**0.14 (0.14, 0.15)**	0.53 (0.52, 0.54)	0.58 (0.57, 0.59)
	20	**0.20 (0.20, 0.20)**	3.4 (3.0, 3.9)	0.54 (0.53, 0.54)	0.58 (0.57, 0.59)
	40	**0.21 (0.20, 0.21)**	43 (5.7, 80)	0.54 (0.54, 0.55)	0.58 (0.57, 0.59)
λ_2	0	0.23 (0.23, 0.24)	**0.11 (0.11, 0.12)**	1.1 (1.1, 1.1)	1.4 (1.4, 1.4)
	20	**0.25 (0.25, 0.26)**	4.5 (3.9, 5.2)	1.1 (1.1, 1.1)	1.4 (1.4, 1.4)
	40	**0.26 (0.26, 0.27)**	29 (11, 46)	1.1 (1.1, 1.1)	1.4 (1.4, 1.4)
λ_3	0	0.038 (0.037, 0.040)	0.046 (0.044, 0.049)	**0.0040 (0.0039, 0.0041)**	0.10 (0.10, 0.11)
	20	0.047 (0.046, 0.049)	1.8 (1.1, 2.5)	**0.020 (0.019, 0.020)**	0.10 (0.10, 0.11)
	40	0.050 (0.048, 0.051)	7.6 (5.3, 9.7)	**0.030 (0.029, 0.031)**	0.10 (0.10, 0.11)
λ_4	0	0.049 (0.048, 0.050)	**0.045 (0.044, 0.046)**	0.20 (0.19, 0.20)	0.20 (0.19, 0.20)
	20	**0.060 (0.059, 0.062)**	3.9 (0.66, 7.1)	0.20 (0.19, 0.20)	0.20 (0.19, 0.20)
	40	**0.069 (0.067, 0.070)**	5.5 (4.3, 6.7)	0.20 (0.20, 0.21)	0.20 (0.19, 0.20)

RMSE error with 95% confidence interval.

RMSE error

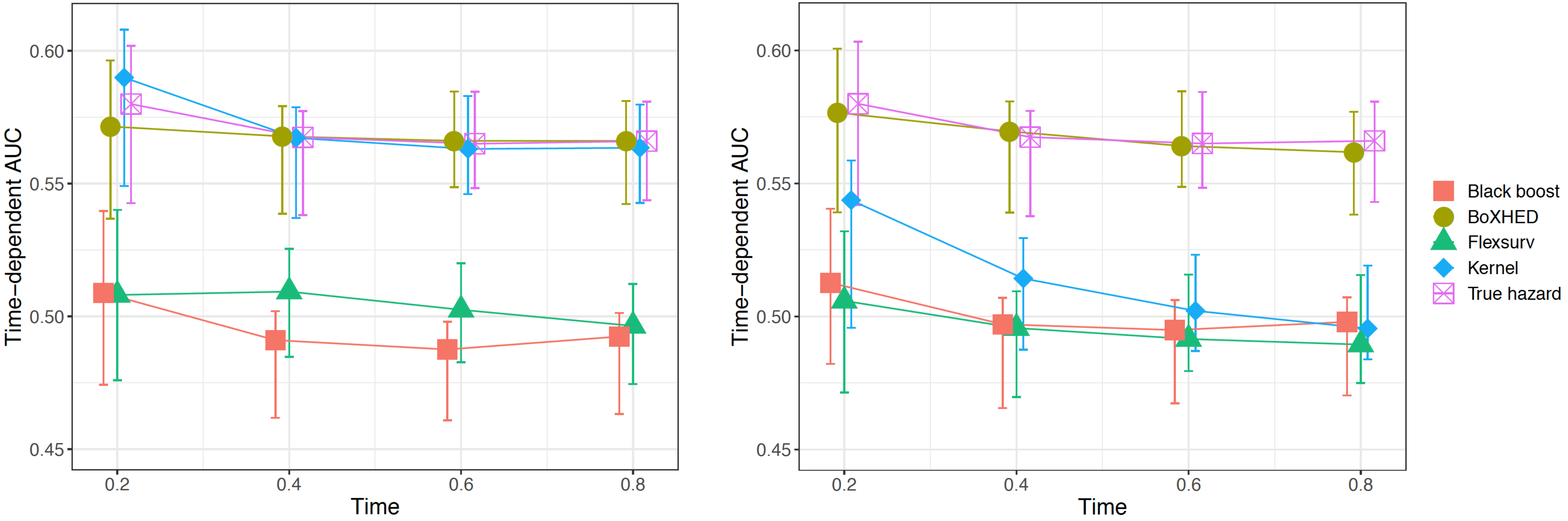
The kernel function is a beta density, resembling λ_1 and λ_2.

Hazard	#Irrelevant covariates	Estimator BoXHED	kernel	flexsurv	blackboost
λ_1	0	0.17 (0.17, 0.17)	**0.14 (0.14, 0.15)**	0.53 (0.52, 0.54)	0.58 (0.57, 0.59)
	20	**0.20 (0.20, 0.20)**	3.4 (3.0, 3.9)	0.54 (0.53, 0.54)	0.58 (0.57, 0.59)
	40	**0.21 (0.20, 0.21)**	43 (5.7, 80)	0.54 (0.54, 0.55)	0.58 (0.57, 0.59)
λ_2	0	0.23 (0.23, 0.24)	**0.11 (0.11, 0.12)**	1.1 (1.1, 1.1)	1.4 (1.4, 1.4)
	20	**0.25 (0.25, 0.26)**	4.5 (3.9, 5.2)	1.1 (1.1, 1.1)	1.4 (1.4, 1.4)
	40	**0.26 (0.26, 0.27)**	29 (11, 46)	1.1 (1.1, 1.1)	1.4 (1.4, 1.4)
λ_3	0	0.038 (0.037, 0.040)	0.046 (0.044, 0.049)	**0.0040 (0.0039, 0.0041)**	0.10 (0.10, 0.11)
	20	0.047 (0.046, 0.049)	1.8 (1.1, 2.5)	**0.020 (0.019, 0.020)**	0.10 (0.10, 0.11)
	40	0.050 (0.048, 0.051)	7.6 (5.3, 9.7)	**0.030 (0.029, 0.031)**	0.10 (0.10, 0.11)
λ_4	0	0.049 (0.048, 0.050)	**0.045 (0.044, 0.046)**	0.20 (0.19, 0.20)	0.20 (0.19, 0.20)
	20	**0.060 (0.059, 0.062)**	3.9 (0.66, 7.1)	0.20 (0.19, 0.20)	0.20 (0.19, 0.20)
	40	**0.069 (0.067, 0.070)**	5.5 (4.3, 6.7)	0.20 (0.20, 0.21)	0.20 (0.19, 0.20)

RMSE error with 95% confidence interval.

flexsurv is correctly specified for λ_3 (log-normal distribution)

Time-dependent AUC



AUC versus time t for the estimators when applied to data simulated from λ_1. Larger AUC values are better. Left: No irrelevant covariates; right: 20 irrelevant covariates.

Framingham heart study data

- 9,697 participants enrolled by 1975 with event follow-up through 2017.
- Many features were measured repeatedly in physical exams almost every two years.
- **Risk factors:** age, gender, systolic blood pressure (SBP), diastolic blood pressure (DBP), total cholesterol (TC), smoking, diabetes, and BMI.
- **Outcome:** first occurrence of a CVD event.

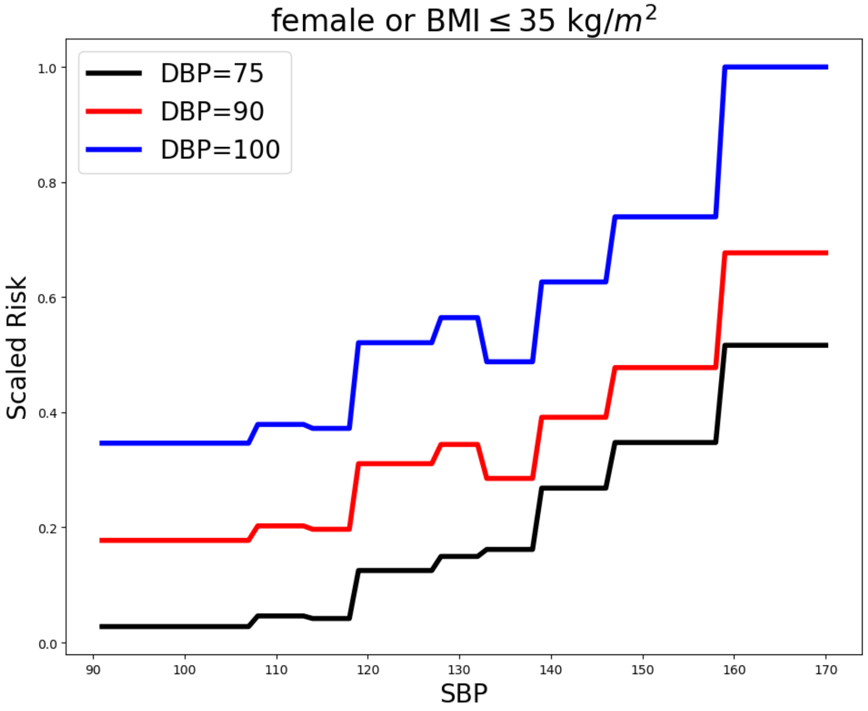
Relationship between SBP and CVD

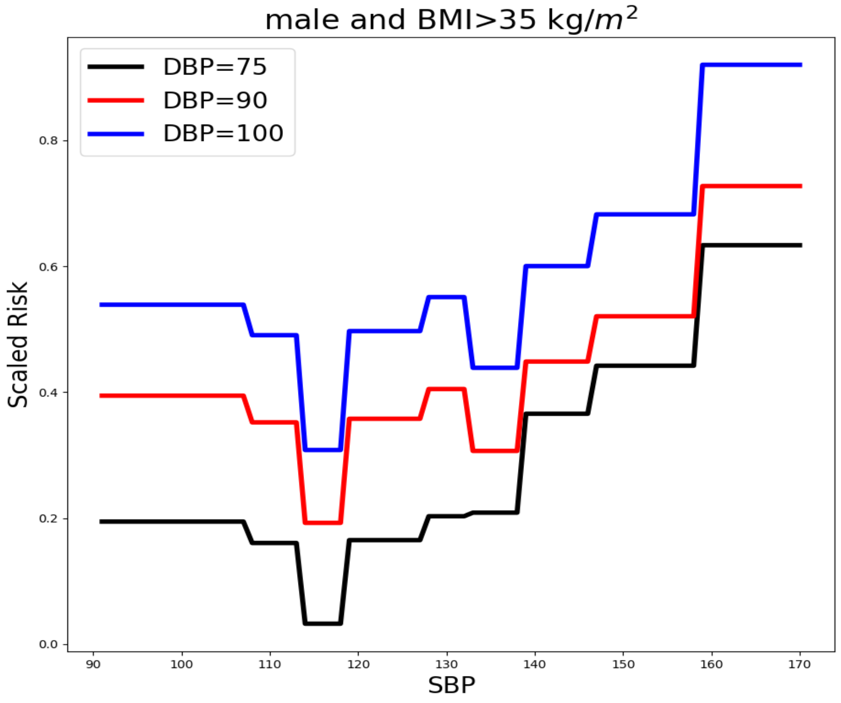
Conflicting clinical literature on how SBP affects CVD risk.

- CVD risk increases with SBP;
- CVD risk decreases with SBP, and then increases (U-shaped);
- some more complicated interaction patterns ...

BoXHED identified novel interaction effects that may partially explain these conflicting findings.

Estimated hazard by SBP





Novel clinical finding

- Hypotheses: The interaction effects SBP×BMI and SBP×Gender are responsible for the reported clinical findings on SBP and CVD risk.

- Validation: SBP×BMI interaction effect is validated using the conventional odds ratio analyses.

Conclusions

- BoXHED is first publicly available software for boosted hazard estimation that is
  - completely nonparametric
  - able to handle time-dependent features
  - applicable to high-dimensional data

- Uncovered a novel interaction effect that may explain conflicting findings on CVD risk in clinical literature.

https://github.com/BoXHED

Q&A