GPSRL: Learning Semi-Parametric Bayesian Survival Rule Lists from Heterogeneous Patient Data

Ameer Hamza Shakur[1], Xiaoning Qian[2], Zhangyang Wang[3], Bobak Mortazavi[2], Shuai Huang[1]

[1]University of Washington, Seattle, WA
[2]Texas A&M University, College Station, TX
[3]The University of Texas at Austin, Austin, TX

Motivation

- Patient health data is often collected from a *heterogeneous* population of patients

Motivation

- Patient health data is often collected from a *heterogeneous* population of patients
- Standard survival models focus on average effect of the covariates on survival outcomes

Motivation

- Patient health data is often collected from a *heterogeneous* population of patients
- Standard survival models focus on average effect of the covariates on survival outcomes
- Advances in sensing technology have provided opportunities to further model *heterogeneity* as well as *non-linearity* of the survival risk.

Survival Analysis

- Cox proportional hazards model:

$$h(t \mid \mathbf{x}_i) = h_0(t)\exp(g(\mathbf{x}_i))$$
$$g(\mathbf{x}_i) = \mathbf{w}'\mathbf{x}_i$$

- Accelerated failure time (AFT) model:

$$T_i = \exp(g(\mathbf{x}_i))T_0$$
$$\log T_i = g(\mathbf{x}_i) + \log(T_0) = g(\mathbf{x}_i) + \epsilon$$
$$\text{assigning a distribution to } \epsilon,$$
$$\epsilon \sim \text{logistic}(\beta^{-1}) \implies T_i \sim \text{log-logistic}(\alpha_i, \beta)$$
$$\text{where, } \alpha_i = \exp(g(\mathbf{x}_i)) \quad g(\mathbf{x}_i) = \mathbf{w}'\mathbf{x}_i$$

Survival Analysis

- Cox proportional hazards model:

$$h(t \mid \mathbf{x}_i) = h_0(t)\exp(g(\mathbf{x}_i))$$
$$g(\mathbf{x}_i) = \mathbf{w}'\mathbf{x}_i$$

Linear effects

- Accelerated failure time (AFT) model:

$$T_i = \exp(g(\mathbf{x}_i))T_0$$
$$\log T_i = g(\mathbf{x}_i) + \log(T_0) = g(\mathbf{x}_i) + \epsilon$$
assigning a distribution to ϵ,
$$\epsilon \sim \text{logistic}(\beta^{-1}) \implies T_i \sim \text{log-logistic}(\alpha_i, \beta)$$
where, $\alpha_i = \exp(g(\mathbf{x}_i))$ $\quad g(\mathbf{x}_i) = \mathbf{w}'\mathbf{x}_i$

Gaussian Processes for Survival Analysis

- Nonlinear effects?
- Replace linear effects $g(\boldsymbol{x})$ with a GP $\mathbf{f}$

Gaussian Processes for Survival Analysis

- Nonlinear effects?
- Replace linear effects *g(**x**)* with a GP **f**
- *GP-Cox*
- **GP-AFT survival model**

$$\boldsymbol{\alpha} = \exp(\mathbf{f}) \text{ where } \mathbf{f} \sim GP(0, \mathbf{K})$$

Gaussian Processes for Survival Analysis

- Nonlinear effects?
- Replace linear effects *g(**x**)* with a GP **f**
- *GP-Cox*
- **GP-AFT survival model**

$$\boldsymbol{\alpha} = \exp(\mathbf{f}) \text{ where } \mathbf{f} \sim GP(0, \mathbf{K}) \qquad \log(\beta) \sim U(0, s)$$

Gaussian Processes for Survival Analysis

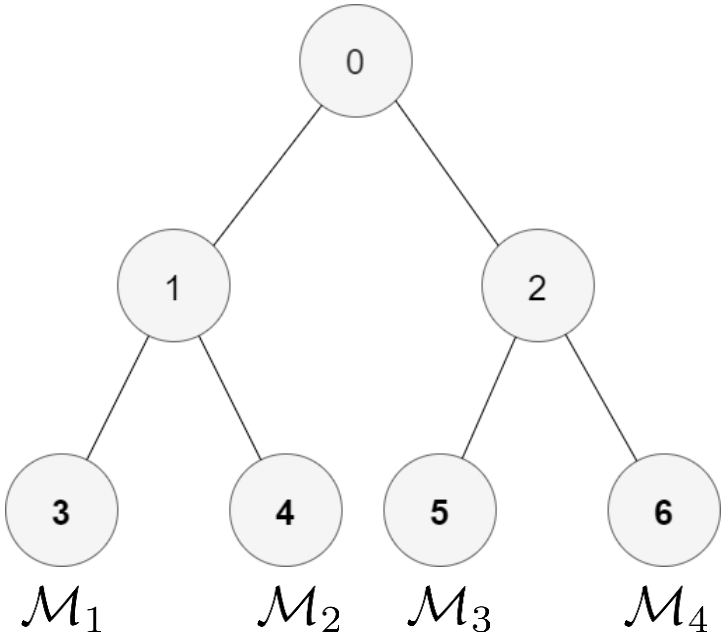
- Nonlinear effects?
- Replace linear effects *g(**x**)* with a GP **f**
- *GP-Cox*
- **GP-AFT survival model**

$$\boldsymbol{\alpha} = \exp(\mathbf{f}) \text{ where } \mathbf{f} \sim GP(0, \mathbf{K}) \qquad \log(\beta) \sim U(0, s)$$

- but, like AFT, β is assumed to be constant and does not depend on co-variates
- restrictive assumption on *shape parameter* in heterogenous datasets

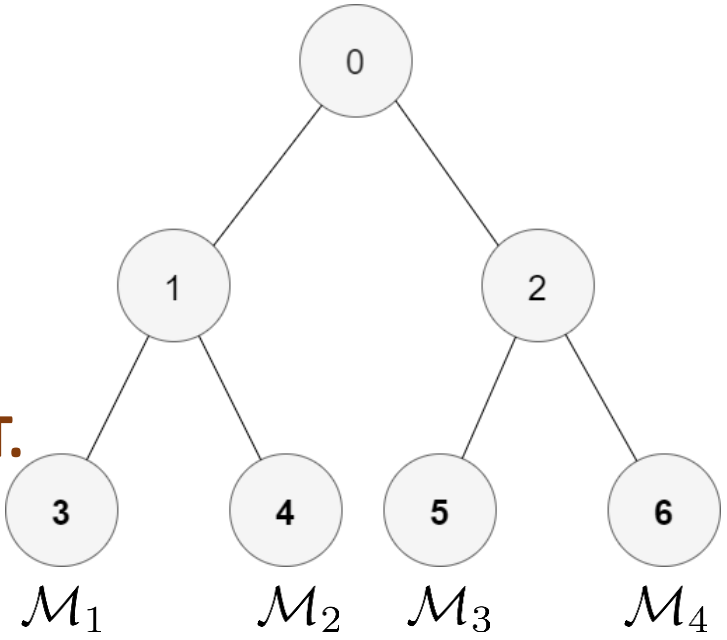
Heterogeneity in GP's

- Data partitioning approaches
  - Ex. Bayesian treed partitioning, Voronoi tessellations etc.
- Trees or more complex structures used for partitioning & modelling heterogeneity



Heterogeneity in GP's

- Data partitioning approaches
  - Ex. Bayesian treed partitioning, Voronoi tessellations etc.
- Trees or more complex structures used for partitioning & modelling heterogeneity
- Fully probabilistic and computationally demanding
- Do not address nonlinear survival models such as **GP-AFT.**



GPSRL: Model

- Pre-mine: R, *a* rule set containing K rules

GPSRL: Model

- Pre-mine: R, *a* rule set containing K rules
- Construct: $d = \{r_1, r_2 \cdots, r_m\} \subset R$ *an ordered rule list* of size *m*

d

$$
\begin{aligned}
&\textit{if } r_1 \textit{ then } t_1 \sim LL(\boldsymbol{\alpha_1}, \beta_1)\\
&\textit{else if } r_2 \textit{ then } t_2 \sim LL(\boldsymbol{\alpha_2}, \beta_2)\\
&\qquad\qquad .\\
&\qquad\qquad .\\
&\textit{else if } r_m \textit{ then } t_m \sim LL(\boldsymbol{\alpha_m}, \beta_m)\\
&\qquad\qquad \textit{else } t_0 \sim LL(\boldsymbol{\alpha_0}, \beta_0)
\end{aligned}
$$

GPSRL: Model

- Pre-mine: R, *a* rule set containing K rules
- Construct: $d = \{r_1, r_2 \cdots, r_m\} \subset R$ *an ordered rule list* of size *m*

d

$$
\begin{aligned}
&\textit{if } r_1 \textit{ then } t_1 \sim LL(\boldsymbol{\alpha_1}, \beta_1) \\
&\textit{else if } r_2 \textit{ then } t_2 \sim LL(\boldsymbol{\alpha_2}, \beta_2) \\
&\qquad\qquad . \\
&\qquad\qquad . \\
&\textit{else if } r_m \textit{ then } t_m \sim LL(\boldsymbol{\alpha_m}, \beta_m) \\
&\qquad\qquad \textit{else } t_0 \sim LL(\boldsymbol{\alpha_0}, \beta_0)
\end{aligned}
$$

- Fit a GP-AFT model at each partition

GPSRL: Model

- Pre-mine: R, *a* rule set containing K rules
- Construct: $d = \{r_1, r_2 \cdots, r_m\} \subset R$ *an ordered rule list* of size *m*

$$
d \left\{ \begin{array}{r}
\textit{if } r_1 \textit{ then } t_1 \sim LL(\boldsymbol{\alpha_1}, \beta_1) \\
\textit{else if } r_2 \textit{ then } t_2 \sim LL(\boldsymbol{\alpha_2}, \beta_2) \\
\vdots \\
\textit{else if } r_m \textit{ then } t_m \sim LL(\boldsymbol{\alpha_m}, \beta_m) \\
\textit{else } t_0 \sim LL(\boldsymbol{\alpha_0}, \beta_0)
\end{array} \right.
$$

- Latent GP to model non-linearities

- Fit a GP-AFT model at each partition

GPSRL: Model

- Pre-mine: R, *a* rule set containing K rules
- Construct: $d = \{r_1, r_2 \cdots, r_m\} \subset R$ *an ordered rule list* of size *m*

$$
d \left\{ \begin{array}{r}
\textit{if } r_1 \textit{ then } t_1 \sim LL(\boldsymbol{\alpha_1}, \beta_1) \\
\textit{else if } r_2 \textit{ then } t_2 \sim LL(\boldsymbol{\alpha_2}, \beta_2) \\
\vdots \\
\textit{else if } r_m \textit{ then } t_m \sim LL(\boldsymbol{\alpha_m}, \beta_m) \\
\textit{else } t_0 \sim LL(\boldsymbol{\alpha_0}, \beta_0)
\end{array} \right.
$$

- Latent GP to model non-linearities
- Ordered rule lists to model heterogeneity

- Fit a GP-AFT model at each partition

GPSRL: Model

- Pre-mine: R, *a* rule set containing K rules
- Construct: $d = \{r_1, r_2 \cdots, r_m\} \subset R$ *an ordered rule list* of size *m*

$$
d \quad \begin{array}{r} \textit{if } r_1 \textit{ then } t_1 \sim LL(\boldsymbol{\alpha_1}, \beta_1) \\ \textit{else if } r_2 \textit{ then } t_2 \sim LL(\boldsymbol{\alpha_2}, \beta_2) \\ \vdots \\ \textit{else if } r_m \textit{ then } t_m \sim LL(\boldsymbol{\alpha_m}, \beta_m) \\ \textit{else } t_0 \sim LL(\boldsymbol{\alpha_0}, \beta_0) \end{array}
$$

- Latent GP to model non-linearities
- Ordered rule lists to model heterogeneity
- Relaxes assumption on *shape parameter,* β

- Fit a GP-AFT model at each partition

GPSRL: Model

- Pre-mine: R, *a* rule set containing K rules
- Construct: $d = \{r_1, r_2 \cdots, r_m\} \subset R$ *an ordered rule list* of size *m*

$$
d \quad \begin{array}{r}
\textit{if } r_1 \textit{ then } t_1 \sim LL(\boldsymbol{\alpha_1}, \beta_1) \\
\textit{else if } r_2 \textit{ then } t_2 \sim LL(\boldsymbol{\alpha_2}, \beta_2) \\
\vdots \\
\textit{else if } r_m \textit{ then } t_m \sim LL(\boldsymbol{\alpha_m}, \beta_m) \\
\textit{else } t_0 \sim LL(\boldsymbol{\alpha_0}, \beta_0)
\end{array}
$$

- Latent GP to model non-linearities
- Ordered rule lists to model heterogeneity
- Relaxes assumption on *shape parameter,* β
- Rule lists strike a balance between greedy and optimal data partitioning

- Fit a GP-AFT model at each partition

GPSRL: Bayesian Inference

GPSRL: Bayesian Inference

- Posterior probability density

$$p(d \mid \mathbf{X}, \mathbf{y}) \propto p(\mathbf{y} \mid \mathbf{X}, d)p(d).$$

- Prior probability

$$p(d) = p(m \mid \lambda) \prod_{j=1}^{m} p(c_j \mid c_1 \cdots c_{j-1}, \eta) p(r_j \mid r_1, \cdots r_{j-1}, c_j)$$

$$p(m \mid \mathcal{R}, \lambda) = \frac{(\lambda^m / m!)}{\sum_{j=1}^{K} (\lambda^j / j!)}, \quad m = 0, 1, \ldots K$$

$$p(c_j \mid c_1 \cdots c_{j-1}, \eta) = \frac{\eta^{c_j} / c_j!}{\sum_{c \in C_j} \eta^c / c!}$$

$$p(r_j \mid r_1, \cdots r_{j-1}, c_j) = \frac{1}{|\{r_i \mid r_i \in \mathcal{R}_j, c_i = c_j\}|}$$

GPSRL: Bayesian Inference

- Log-logistic likelihood

$$p(y \mid \boldsymbol{\alpha}, \beta) = \prod_{i=1}^{M:\boldsymbol{\delta}=1} \frac{(\frac{\boldsymbol{\alpha}_i}{\beta})(\frac{y_i}{\boldsymbol{\alpha}_i})^{\beta-1}}{(1+\frac{y_i}{\boldsymbol{\alpha}_i}\beta)^2} \prod_{j=1}^{N:\boldsymbol{\delta}_j=0} \frac{1}{1+(\frac{y_i}{\boldsymbol{\alpha}_i})^{\beta}}.$$

- Marginal likelihood
  - Likelihood is not conjugate-Gaussian
  - Laplace approximation

$$q(\mathbf{f}) \approx p(\mathbf{f} \mid \mathbf{y}, \mathbf{X})$$

$$p(\mathbf{y} \mid \mathbf{X}) \approx p_L(\mathbf{y} \mid \mathbf{X}) = \int p(\mathbf{y} \mid \mathbf{f}) q(\mathbf{f}) d\mathbf{f}$$

GPSRL: Bayesian Inference

- Log-logistic likelihood

$$p(y \mid \boldsymbol{\alpha}, \beta) = \prod_{i=1}^{M:\boldsymbol{\delta}=1} \frac{(\frac{\boldsymbol{\alpha}_i}{\beta})(\frac{y_i}{\boldsymbol{\alpha}_i})^{\beta-1}}{(1+\frac{y_i}{\boldsymbol{\alpha}_i}\beta)^2} \prod_{j=1}^{N:\boldsymbol{\delta}_j=0} \frac{1}{1+(\frac{y_i}{\boldsymbol{\alpha}_i})^{\beta}}.$$

- Marginal likelihood
  - Likelihood is not conjugate-Gaussian
  - Laplace approximation

$$q(\mathbf{f}) \approx p(\mathbf{f} \mid \mathbf{y}, \mathbf{X})$$

$$p(\mathbf{y} \mid \mathbf{X}) \approx p_L(\mathbf{y} \mid \mathbf{X}) = \int p(\mathbf{y} \mid \mathbf{f}) q(\mathbf{f}) d\mathbf{f}$$

$$p_L(\mathbf{y} \mid \mathbf{X}, d) = \prod_{j=0}^{m} p_L(\mathbf{y}_j \mid \mathbf{X}_j)$$

Approx. marginal likelihood

GPSRL: MCMC sampling

- Initial random list: $d^0 \sim p(d)$

GPSRL: MCMC sampling

- Initial random list: $d^0 \sim p(d)$
- At step *t* in the sequence: $d^{t+1} \sim Q(d^t)$
- Proposal distribution: $$Q(d^{t+1} \mid d^t, \mathcal{R}) = \begin{cases} \frac{1}{(K-m^t)(m^t+1)} & \text{if a rule is added} \\ \frac{1}{m^t} & \text{if a rule is removed} \\ \frac{1}{m^t(m^t-1)} & \text{if a rule is moved.} \end{cases}$$

GPSRL: MCMC sampling

- Initial random list: $d^0 \sim p(d)$
- At step *t* in the sequence: $d^{t+1} \sim Q(d^t)$
- Proposal distribution: $$Q(d^{t+1} \mid d^t, \mathcal{R}) = \begin{cases} \frac{1}{(K-m^t)(m^t+1)} & \text{if a rule is added} \\ \frac{1}{m^t} & \text{if a rule is removed} \\ \frac{1}{m^t(m^t-1)} & \text{if a rule is moved.} \end{cases}$$
- Acceptance probability: $$\pi(d^{t+1} \mid d^t) = \min\left\{ \frac{Q(d^{t+1}, d^t)}{Q(d^t, d^{t+1})} \frac{p_L(\mathbf{y} \mid \mathbf{X}, d^{t+1}) p(d^{t+1})}{p_L(\mathbf{y} \mid \mathbf{X}, d^t) p(d^t)}, 1 \right\}.$$

GPSRL: MCMC sampling

- Initial random list: $d^0 \sim p(d)$
- At step *t* in the sequence: $d^{t+1} \sim Q(d^t)$
- Proposal distribution: $$Q(d^{t+1} \mid d^t, \mathcal{R}) = \begin{cases} \frac{1}{(K-m^t)(m^t+1)} & \text{if a rule is added} \\ \frac{1}{m^t} & \text{if a rule is removed} \\ \frac{1}{m^t(m^t-1)} & \text{if a rule is moved.} \end{cases}$$
- Acceptance probability: $$\pi(d^{t+1} \mid d^t) = \min\left\{ \frac{Q(d^{t+1}, d^t)}{Q(d^t, d^{t+1})} \frac{p_L(\mathbf{y} \mid \mathbf{X}, d^{t+1}) p(d^{t+1})}{p_L(\mathbf{y} \mid \mathbf{X}, d^t) p(d^t)}, 1 \right\}.$$
- MCMC sequence $d^0, d^1 \cdots d^n \to p(d \mid \mathbf{X}, \mathbf{y})$

Experiments – Performance evaluation

- Two datasets
  - Synthetic
  - MIMIC-III Sepsis

1. GPy: A Gaussian process framework in Python," http://github.com/SheffieldML/GPy

Experiments – Performance evaluation

- Two datasets
  - Synthetic
  - MIMIC-III Sepsis
- Performance evaluation:
  - Concordance Index (C-Index)
  - Negative log probability density (NLPD)

$$\mathrm{NLPD}(\mathbf{y}^*, \mathbf{X}^*) = \frac{\sum_{i=1}^{N} p(y_i^* | \mathbf{x}_i^*, \mathbf{X}, \mathbf{y})}{N}$$

1. GPy: A Gaussian process framework in Python," http://github.com/SheffieldML/GPy

Experiments – Performance evaluation

- Two datasets
  - Synthetic
  - MIMIC-III Sepsis
- Performance evaluation:
  - Concordance Index (C-Index)
  - Negative log probability density (NLPD)

$$\mathrm{NLPD}(\mathbf{y}^*, \mathbf{X}^*) = \frac{\sum_{i=1}^{N} p(y_i^* | \mathbf{x}_i^*, \mathbf{X}, \mathbf{y})}{N}$$

1. GPy: A Gaussian process framework in Python," http://github.com/SheffieldML/GPy

Results – Synthetic Data

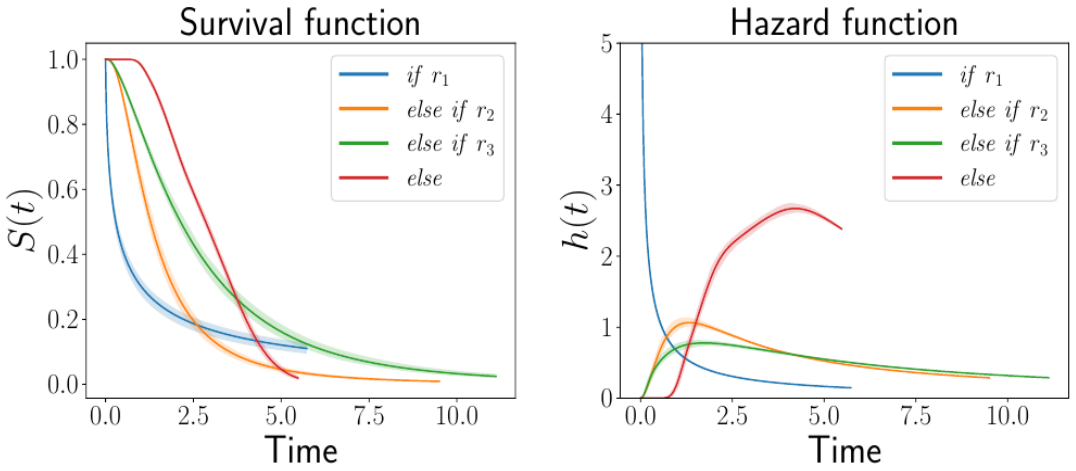
Table 1: Estimate of ordered rule list d from the posterior

Rules	
r_1	$x_3 \leq 0.259$
r_2	$x_4 > 0.596$ & $x_3 > 0.196$
r_3	$x_4 \leq 0.677$ & $x_4 > 0.192$

Results – Synthetic Data

Table 1: Estimate of ordered rule list d from the posterior

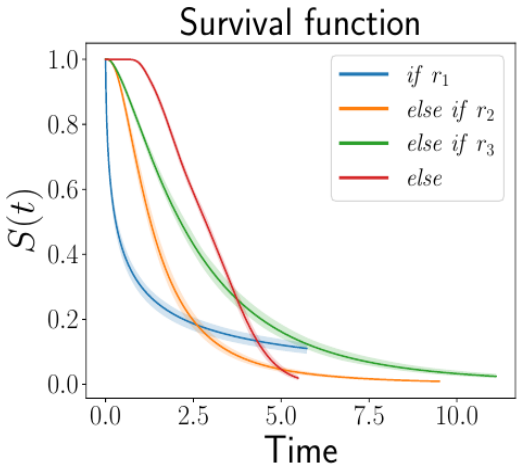
Rules	
r_1	$x_3 \leq 0.259$
r_2	$x_4 > 0.596$ & $x_3 > 0.196$
r_3	$x_4 \leq 0.677$ & $x_4 > 0.192$

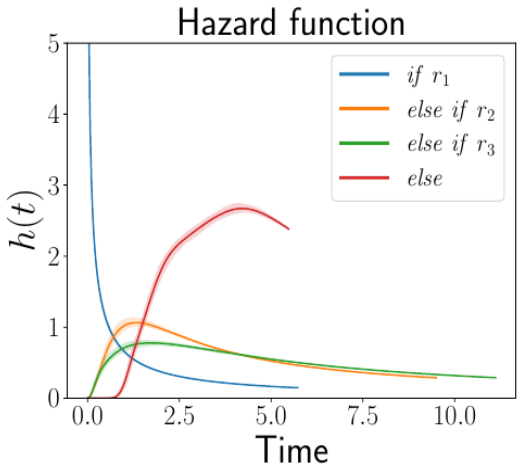


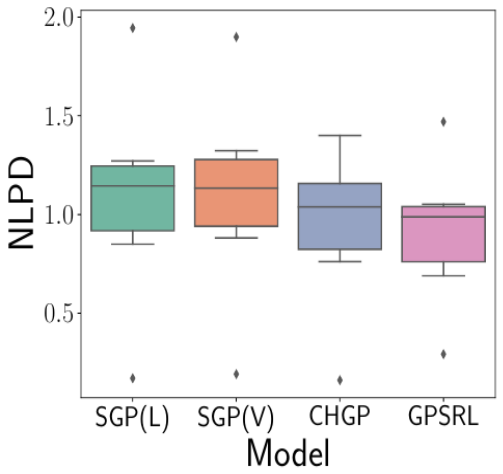
Results – Synthetic Data

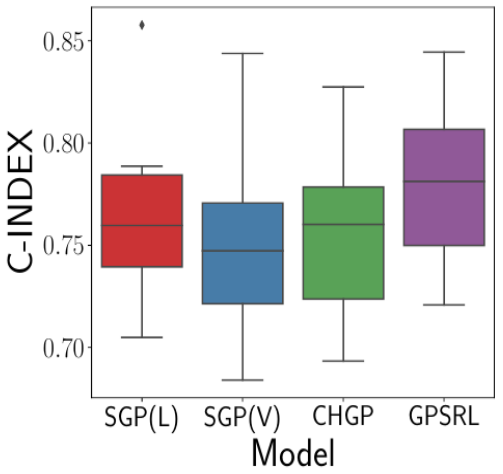
Table 1: Estimate of ordered rule list d from the posterior

Rules	
r_1	$x_3 \leq 0.259$
r_2	$x_4 > 0.596$ & $x_3 > 0.196$
r_3	$x_4 \leq 0.677$ & $x_4 > 0.192$









Experiments – MIMIC III

- MIMIC-III, a database comprising anonymized information relating to patients admitted to BIDMC[2] during 2001-02 and diagnosed with sepsis.

Table 1: Estimate of ordered rule list d from the posterior

Rules	
r_1	artpH-(mean) <= 7.249
r_2	O2sat-(sd) <= 4.66 & diaBP-(mean) <= 61.26
r_3	O2sat-(sd) <= 4.8

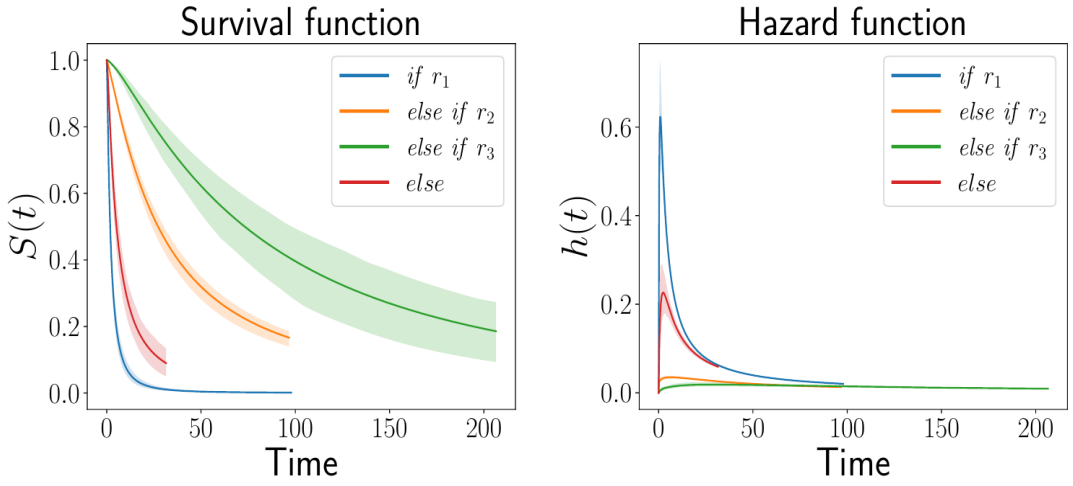
2. Beth Israel Beth Israel Deaconess Medical Center (BIDMC), Boston, MA

Experiments – MIMIC III

- MIMIC-III, a database comprising anonymized information relating to patients admitted to BIDMC[2] during 2001-02 and diagnosed with sepsis.

Table 1: Estimate of ordered rule list d from the posterior

Rules	
r_1	artpH-(mean) <= 7.249
r_2	O2sat-(sd) <= 4.66 & diaBP-(mean) <= 61.26
r_3	O2sat-(sd) <= 4.8



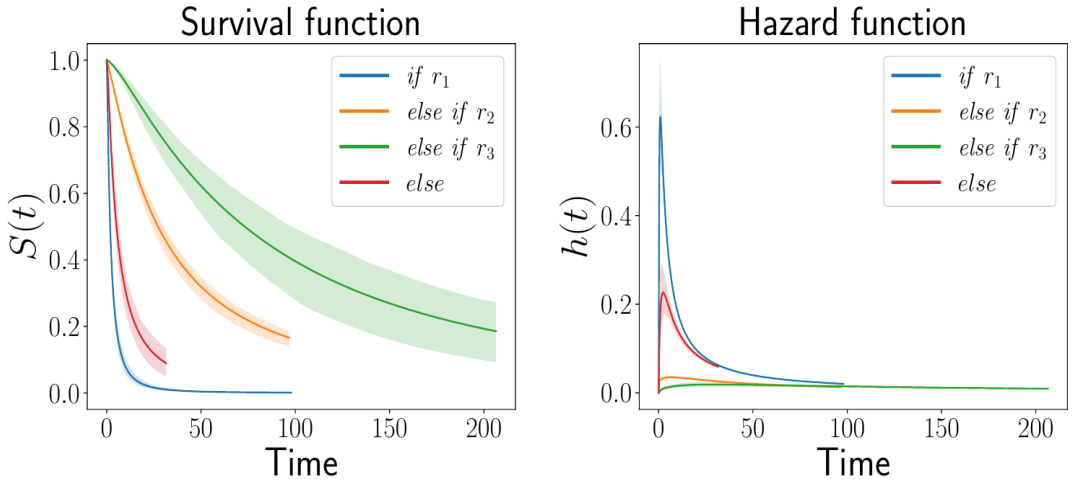
2. Beth Israel Beth Israel Deaconess Medical Center (BIDMC), Boston, MA

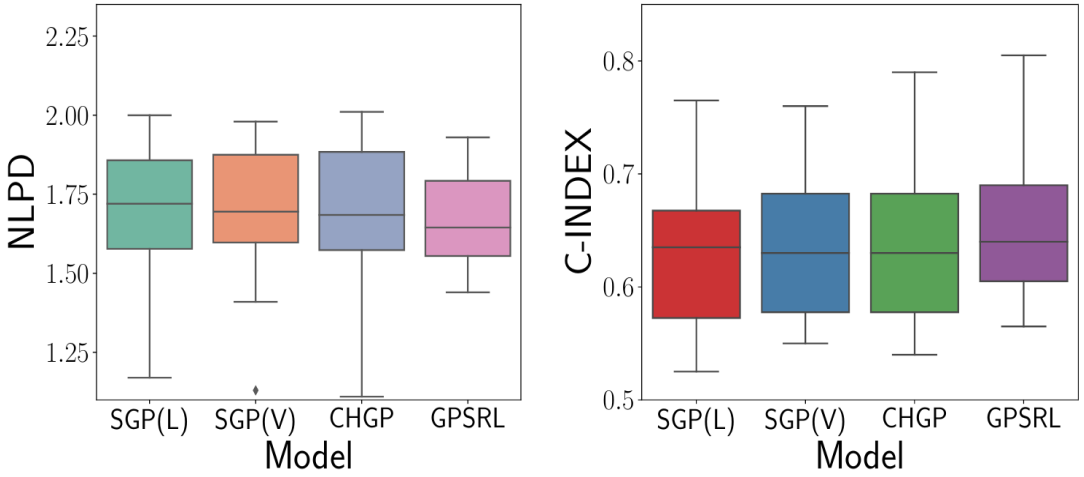
Experiments – MIMIC III

- MIMIC-III, a database comprising anonymized information relating to patients admitted to BIDMC[2] during 2001-02 and diagnosed with sepsis.

Table 1: Estimate of ordered rule list d from the posterior

Rules	
r_1	artpH-(mean) <= 7.249
r_2	O2sat-(sd) <= 4.66 & diaBP-(mean) <= 61.26
r_3	O2sat-(sd) <= 4.8





2. Beth Israel Beth Israel Deaconess Medical Center (BIDMC), Boston, MA

Summary

Summary

- Gaussian Process Survival Rule Lists (GPSRL) to model heterogeneity in survival data-sets

Summary

- Gaussian Process Survival Rule Lists (GPSRL) to model heterogeneity in survival data-sets
- Semi-parametric Bayesian framework to partition the data into subsets with different survival characteristics.

Summary

- Gaussian Process Survival Rule Lists (GPSRL) to model heterogeneity in survival data-sets
- Semi-parametric Bayesian framework to partition the data into subsets with different survival characteristics.
- Addresses some limitations of standard survival Gaussian process models

Summary

- Gaussian Process Survival Rule Lists (GPSRL) to model heterogeneity in survival data-sets
- Semi-parametric Bayesian framework to partition the data into subsets with different survival characteristics.
- Addresses some limitations of standard survival Gaussian process models
- Interpretability in the form of rules

Summary

- Gaussian Process Survival Rule Lists (GPSRL) to model heterogeneity in survival data-sets
- Semi-parametric Bayesian framework to partition the data into subsets with different survival characteristics.
- Addresses some limitations of standard survival Gaussian process models
- Interpretability in the form of rules
- Performance evaluations demonstrate the effectiveness of our model

Thank You!

Synthetic data simulation

- Simulated a heterogeneous survival dataset of size N = 1000, P = 4

$$D = (\mathbf{t}, \boldsymbol{\delta}, \mathbf{X}) \qquad (\mathbf{x}_i \sim U(0,1)\ \forall i \in 1:N)$$

- Event times sampled from log-logistic (*LL)* distribution

$$t_i \sim LL(\alpha_i, \beta_i)$$

$$\alpha(\mathbf{x}) = I_1\alpha_1(\mathbf{x}) + I_2\alpha_2(\mathbf{x}) + I_3\alpha_3(\mathbf{x})$$
$$\beta(\mathbf{x}) = I_1\beta_1(\mathbf{x}) + I_2\beta_2(\mathbf{x}) + I_3\beta_3(\mathbf{x})$$

$$\alpha_i(\mathbf{x}) = a_1 \exp\Big(a_2\big(\sum_{k=1}^{2} \exp(a_3(x[k] - a_4)^2)\big) + \sum_{k=3}^{4} \sin(\pi \mathbf{x}[k]^2)\Big)$$

$$\beta_i(\mathbf{x}) = b_1 \exp\Big(\sum_{k=1}^{2} \sin(2\pi \mathbf{x}[k]^2) + \sum_{k=3}^{4} \cos(2\pi \mathbf{x}[k]^2)\Big),$$