High-Dimensional Variable Selection for Competing Risks with Cooperative Penalized Regression

«CooPeR»

Lukas Burk

BIPS — LMU/SLDS — MCML

2023-05-12

Motivation

- Setting: High-dimensional survival data w/ competing risks

Motivation

- Setting: High-dimensional survival data w/ competing risks
  - e.g.: Time to death from cause 1 (bladder cancer) or cause 2 (other)

Motivation

- Setting: High-dimensional survival data w/ competing risks
  - e.g.: Time to death from cause 1 (bladder cancer) or cause 2 (other)
- Typical approach:

Motivation

- Setting: High-dimensional survival data w/ competing risks
  - e.g.: Time to death from cause 1 (bladder cancer) or cause 2 (other)
- Typical approach:
  - Fit cause-specific model(s) for event(s) of interest

Motivation

- Setting: High-dimensional survival data w/ competing risks
  - e.g.: Time to death from cause 1 (bladder cancer) or cause 2 (other)
- Typical approach:
  - Fit cause-specific model(s) for event(s) of interest
  - Treats other events as censored

Motivation

- Setting: High-dimensional survival data w/ competing risks
  - e.g.: Time to death from cause 1 (bladder cancer) or cause 2 (other)
- Typical approach:
  - Fit cause-specific model(s) for event(s) of interest
  - Treats other events as censored
  - $\rightarrow$ discards (potential) shared information

Motivation

- Setting: High-dimensional survival data w/ competing risks
  - e.g.: Time to death from cause 1 (bladder cancer) or cause 2 (other)
- Typical approach:
  - Fit cause-specific model(s) for event(s) of interest
  - Treats other events as censored
  - $\rightarrow$ discards (potential) shared information
- **Main goal**: Fit cause-specific model for event 1 *using shared information* from event 2

Building Blocks

1. Penalized Cox regression

Building Blocks

1. Penalized Cox regression
   - Elastic net / feature-weighted elastic net (fwelnet)

Building Blocks

1. Penalized Cox regression
   - Elastic net / feature-weighted elastic net (fwelnet)
2. An iterative Algorithm

Building Blocks

1. Penalized Cox regression
   - Elastic net / feature-weighted elastic net (fwelnet)
2. An iterative Algorithm
   - Adapted from multi-task algorithm by fwelnet authors

Building Blocks

1. Penalized Cox regression
   - Elastic net / feature-weighted elastic net (fwelnet)
2. An iterative Algorithm
   - Adapted from multi-task algorithm by fwelnet authors
3. Assumption of shared information between causes

Building Blocks

1. Penalized Cox regression
   - Elastic net / feature-weighted elastic net (fwelnet)
2. An iterative Algorithm
   - Adapted from multi-task algorithm by fwelnet authors
3. Assumption of shared information between causes
   - Idea: Some features predictive for event 1 will also be predictive for event 2

Foundation: Elastic Net

The elastic net objective function with some negative log-likelihood term:

$$\underset{\beta}{\operatorname{argmin}} \quad \mathrm{NLL}(\beta) + \lambda \sum_{j=1}^{p} \left(\alpha |\beta_j| + \frac{1-\alpha}{2} \beta_j^2 \right)$$

- $\lambda \in \mathbb{R}_+$ controls overall penalty

Foundation: Elastic Net

The elastic net objective function with some negative log-likelihood term:

$$\underset{\beta}{\operatorname{argmin}} \quad \mathrm{NLL}(\beta) + \lambda \sum_{j=1}^{p} \left(\alpha |\beta_j| + \frac{1-\alpha}{2} \beta_j^2 \right)$$

- $\lambda \in \mathbb{R}_+$ controls overall penalty
- Higher $\to$ larger penalty on all β_j, **equally**

Foundation: Elastic Net

The elastic net objective function with some negative log-likelihood term:

$$\underset{\beta}{\operatorname{argmin}} \quad \mathrm{NLL}(\beta) + \lambda \sum_{j=1}^{p} \left(\alpha |\beta_j| + \frac{1-\alpha}{2} \beta_j^2 \right)$$

- $\lambda \in \mathbb{R}_+$ controls overall penalty
- Higher $\rightarrow$ larger penalty on all β_j, **equally**
- $\alpha \in [0, 1]$ is the mixing parameter

Foundation: Elastic Net

The elastic net objective function with some negative log-likelihood term:

$$\underset{\beta}{\operatorname{argmin}} \quad \mathrm{NLL}(\beta) + \lambda \sum_{j=1}^{p} \left(\alpha |\beta_j| + \frac{1-\alpha}{2} \beta_j^2 \right)$$

- $\lambda \in \mathbb{R}_+$ controls overall penalty
- Higher $\rightarrow$ larger penalty on all β_j, **equally**

- $\alpha \in [0,1]$ is the mixing parameter
- $1 \Rightarrow$ only ℓ_1 penalty (LASSO)

Foundation: Elastic Net

The elastic net objective function with some negative log-likelihood term:

$$\underset{\beta}{\operatorname{argmin}} \quad \mathrm{NLL}(\beta) + \lambda \sum_{j=1}^{p} \left(\alpha |\beta_j| + \frac{1-\alpha}{2} \beta_j^2 \right)$$

- $\lambda \in \mathbb{R}_+$ controls overall penalty
- Higher $\rightarrow$ larger penalty on all β_j, **equally**

- $\alpha \in [0, 1]$ is the mixing parameter
- $1 \Rightarrow$ only ℓ_1 penalty (LASSO)
- $0 \Rightarrow$ only ℓ_2 penalty (ridge)

Elastic Net: Flexibility?

- What if we don't want to penalize all β_j equally?

Elastic Net: Flexibility?

- What if we don't want to penalize all β_j equally?
- This does **not** work:

Elastic Net: Flexibility?

- What if we don't want to penalize all β_j equally?
- This does **not** work:

Elastic Net: Flexibility?

- What if we don't want to penalize all β_j equally?
- This does **not** work:

$$\underset{\beta}{\operatorname{argmin}} \quad \mathrm{NLL}(\beta) + \sum_{j=1}^{p} \lambda_p \left(\alpha |\beta_j| + \frac{1-\alpha}{2} \beta_j^2 \right)$$

Elastic Net: Flexibility?

- What if we don't want to penalize all β_j equally?
- This does **not** work:

$$\underset{\beta}{\operatorname{argmin}} \quad \mathrm{NLL}(\beta) + \sum_{j=1}^{p} \lambda_p \left(\alpha |\beta_j| + \frac{1-\alpha}{2} \beta_j^2 \right)$$

$\rightarrow$ Need different approach

Feature-Weighted Elastic Net (`fwelnet`)

- Motivation: Using external information

Feature-Weighted Elastic Net (`fwelnet`)

- Motivation: Using external information
- Adjust penalization weights on individual or groups of features

Feature-Weighted Elastic Net (`fwelnet`)

- Motivation: Using external information
- Adjust penalization weights on individual or groups of features
- Assign weights / groups via matrix $\mathbf{Z} \in \mathbb{R}^{p \times K}$

Feature-Weighted Elastic Net (`fwelnet`)

- Motivation: Using external information
- Adjust penalization weights on individual or groups of features
- Assign weights / groups via matrix $\mathbf{Z} \in \mathbb{R}^{p \times K}$

Feature-Weighted Elastic Net (`fwelnet`)

- Motivation: Using external information
- Adjust penalization weights on individual or groups of features
- Assign weights / groups via matrix $\mathbf{Z} \in \mathbb{R}^{p \times K}$

Two Applications

1. Assign features to K groups w/ separate penalization weights

(Tay et al. 2023)

Feature-Weighted Elastic Net (`fwelnet`)

- Motivation: Using external information
- Adjust penalization weights on individual or groups of features
- Assign weights / groups via matrix $\mathbf{Z} \in \mathbb{R}^{p \times K}$

Two Applications

1. Assign features to K groups w/ separate penalization weights
2. Adjust penalization weights within group

(Tay et al. 2023)

Feature Weighting: Groups

Example for $p = 5$ features $X_{1,2,3,4,5}$ and $K = 2$ groups

Feature Weighting: Groups

Example for $p = 5$ features $X_{1,2,3,4,5}$ and $K = 2$ groups

$$\mathbf{Z} = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}$$

- X_1, X_2 group 1

Feature Weighting: Groups

Example for $p = 5$ features $X_{1,2,3,4,5}$ and $K = 2$ groups

$$\mathbf{Z} = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}$$

- X_1, X_2 group 1
- X_3, X_4, X_5 group 2

Feature Weighting: Groups

Example for $p = 5$ features $X_{1,2,3,4,5}$ and $K = 2$ groups

$$\mathbf{Z} = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}$$

- X_1, X_2 group 1
- X_3, X_4, X_5 group 2

Feature Weighting: Groups

Example for $p = 5$ features $X_{1,2,3,4,5}$ and $K = 2$ groups

$$\mathbf{Z} = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}$$

- X_1, X_2 group 1
- X_3, X_4, X_5 group 2

Interesting when e.g. $p >> 1000$ w/ 50 clinical + 5000 gene expression features

Feature Weighting: Per Variable

Example for $p = 4$ features $X_{1,2,3,4}$:

Feature Weighting: Per Variable

Example for $p = 4$ features $X_{1,2,3,4}$:

$$\mathbf{Z} = \begin{pmatrix} 0.5 \\ 1 \\ 2 \\ 0 \end{pmatrix}$$

- X_1: Less important, strong penalization

Feature Weighting: Per Variable

Example for $p = 4$ features $X_{1,2,3,4}$:

$$\mathbf{Z} = \begin{pmatrix} 0.5 \\ 1 \\ 2 \\ 0 \end{pmatrix}$$

- X_1: Less important, strong penalization
- X_2, X_3: More important $\rightarrow$ weaker penalization

Feature Weighting: Per Variable

Example for $p = 4$ features $X_{1,2,3,4}$:

$$\mathbf{z} = \begin{pmatrix} 0.5 \\ 1 \\ 2 \\ 0 \end{pmatrix}$$

- X_1: Less important, strong penalization
- X_2, X_3: More important $\rightarrow$ weaker penalization
- X_4: "Irrelevant" $\rightarrow$ stronger penalization

Feature-Weighting: New Objective Function

$$\underset{\beta}{\operatorname{argmin}} \quad \mathrm{NLL}(\beta) + \lambda \sum_{j=1}^{p} w_j(\theta) \left(\alpha |\beta_j| + \frac{1-\alpha}{2} \beta_j^2 \right)$$

Feature-Weighting: New Objective Function

$$\underset{\beta}{\operatorname{argmin}} \quad \mathrm{NLL}(\beta) + \lambda \sum_{j=1}^{p} \color{blue}{w_j(\theta)} \color{black}\left(\alpha |\beta_j| + \frac{1-\alpha}{2} \beta_j^2 \right)$$

$$\color{blue}{w_j(\theta)} \color{black}= \frac{\sum_{l=1}^{p} \exp(\mathbf{z}_l^T \theta)}{p \exp(\mathbf{z}_j^T \theta)}$$

- **Penalization weight** of β_j based on corresponding value in $\mathbf{Z}$ and parameter $\theta \in \mathbb{R}^{K \times 1}$

Penalization Weights

- $w_j(\theta)$ is chosen heuristically by the authors for desirable properties

Penalization Weights

- $w_j(\theta)$ is chosen heuristically by the authors for desirable properties
- $\mathbf{z}_j^T\theta$ acts as a *score*

Penalization Weights

- $w_j(\theta)$ is chosen heuristically by the authors for desirable properties
- $\mathbf{z}_j^T \theta$ acts as a *score*
  - $= 0$, reduces to original elastic net

Penalization Weights

- $w_j(\theta)$ is chosen heuristically by the authors for desirable properties
- $\mathbf{z}_j^T\theta$ acts as a *score*
  - $= 0$, reduces to original elastic net
  - Higher score $\rightarrow$ lower w_j, feature is "more important"

Feature-Weighting: Single Group

- $\mathbf{Z} \in \mathbb{R}^{p \times 1}$: No groups, just weights

Feature-Weighting: Single Group

- $\mathbf{Z} \in \mathbb{R}^{p \times 1}$: No groups, just weights
- Simulation from Tay et al. (2023): $\mathbf{Z}$ set to noisy version of true $|\beta|$

Feature-Weighting: Single Group

- $\mathbf{Z} \in \mathbb{R}^{p \times 1}$: No groups, just weights
- Simulation from Tay et al. (2023): $\mathbf{Z}$ set to noisy version of true $|\beta|$

Feature-Weighting: Single Group

- $\mathbf{Z} \in \mathbb{R}^{p \times 1}$: No groups, just weights
- Simulation from Tay et al. (2023): $\mathbf{Z}$ set to noisy version of true $|\beta|$
- Larger $|\beta_j| \Rightarrow$ weaker penalization for $\hat{\beta}_j$

Feature-Weighting: Single Group

- $\mathbf{Z} \in \mathbb{R}^{p \times 1}$: No groups, just weights
- Simulation from Tay et al. (2023): $\mathbf{Z}$ set to noisy version of true $|\beta|$
- Larger $|\beta_j| \Rightarrow$ weaker penalization for $\hat{\beta}_j$
- $|\beta_j| \approx 0 \Rightarrow$ stronger penalization for $\hat{\beta}_j$

Application for Multi-Task Learning

- Authors suggest multi-task learning algorithm: $\mathbf{X}$ and targets $\mathbf{y}_1, \mathbf{y}_2$

Application for Multi-Task Learning

- Authors suggest multi-task learning algorithm: $\mathbf{X}$ and targets $\mathbf{y}_1, \mathbf{y}_2$

Application for Multi-Task Learning

- Authors suggest multi-task learning algorithm: $\mathbf{X}$ and targets $\mathbf{y}_1, \mathbf{y}_2$

1. Set $\beta_1^{(0)}, \beta_2^{(0)}$ to `glmnet` solution for $(\mathbf{X}, \mathbf{y}_1), (\mathbf{X}, \mathbf{y}_2)$ respectively

Application for Multi-Task Learning

- Authors suggest multi-task learning algorithm: $\mathbf{X}$ and targets $\mathbf{y}_1, \mathbf{y}_2$

1. Set $\beta_1^{(0)}, \beta_2^{(0)}$ to `glmnet` solution for $(\mathbf{X}, \mathbf{y}_1), (\mathbf{X}, \mathbf{y}_2)$ respectively
2. For $k = 0, 1, \ldots$:

Application for Multi-Task Learning

- Authors suggest multi-task learning algorithm: $\mathbf{X}$ and targets $\mathbf{y}_1, \mathbf{y}_2$

1. Set $\beta_1^{(0)}, \beta_2^{(0)}$ to `glmnet` solution for $(\mathbf{X}, \mathbf{y}_1), (\mathbf{X}, \mathbf{y}_2)$ respectively
2. For $k = 0, 1, ...$:
   a) $\mathbf{Z}_2 = \left|\beta_1^{(k)}\right|$. Fit `fwelnet` with $(\mathbf{X}, \mathbf{y}_2, \mathbf{Z}_2)$

Application for Multi-Task Learning

- Authors suggest multi-task learning algorithm: $\mathbf{X}$ and targets $\mathbf{y}_1, \mathbf{y}_2$

1. Set $\beta_1^{(0)}, \beta_2^{(0)}$ to `glmnet` solution for $(\mathbf{X}, \mathbf{y}_1), (\mathbf{X}, \mathbf{y}_2)$ respectively
2. For $k = 0, 1, \ldots$:
   a) $\mathbf{Z}_2 = \left|\beta_1^{(k)}\right|$. Fit `fwelnet` with $(\mathbf{X}, \mathbf{y}_2, \mathbf{Z}_2)$
      - Set $\left|\beta_2^{(k+1)}\right|$ to solution with optimal `lambda`

Application for Multi-Task Learning

- Authors suggest multi-task learning algorithm: $\mathbf{X}$ and targets $\mathbf{y}_1, \mathbf{y}_2$

1. Set $\beta_1^{(0)}, \beta_2^{(0)}$ to `glmnet` solution for $(\mathbf{X}, \mathbf{y}_1), (\mathbf{X}, \mathbf{y}_2)$ respectively
2. For $k = 0, 1, \ldots$:
   a) $\mathbf{Z}_2 = \left|\beta_1^{(k)}\right|$. Fit `fwelnet` with $(\mathbf{X}, \mathbf{y}_2, \mathbf{Z}_2)$
      - Set $\left|\beta_2^{(k+1)}\right|$ to solution with optimal `lambda`

   b) $\mathbf{Z}_1 = \left|\beta_2^{(k+1)}\right|$. Fit `fwelnet` with $(\mathbf{X}, \mathbf{y}_1, \mathbf{Z}_1)$

Application for Multi-Task Learning

- Authors suggest multi-task learning algorithm: $\mathbf{X}$ and targets $\mathbf{y}_1, \mathbf{y}_2$

1. Set $\beta_1^{(0)}, \beta_2^{(0)}$ to `glmnet` solution for $(\mathbf{X}, \mathbf{y}_1), (\mathbf{X}, \mathbf{y}_2)$ respectively
2. For $k = 0, 1, \ldots$:
    a) $\mathbf{Z}_2 = \left|\beta_1^{(k)}\right|$. Fit `fwelnet` with $(\mathbf{X}, \mathbf{y}_2, \mathbf{Z}_2)$
        - Set $\left|\beta_2^{(k+1)}\right|$ to solution with optimal `lambda`

    b) $\mathbf{Z}_1 = \left|\beta_2^{(k+1)}\right|$. Fit `fwelnet` with $(\mathbf{X}, \mathbf{y}_1, \mathbf{Z}_1)$
        - Set $\left|\beta_1^{(k+1)}\right|$ to solution with optimal `lambda`

Transfer to Competing Risks

- Setting with two outcomes/event types: $(\mathbf{t}_1, \delta_1), (\mathbf{t}_2, \delta_2)$

Transfer to Competing Risks

- Setting with two outcomes/event types: $(\mathbf{t}_1, \delta_1), (\mathbf{t}_2, \delta_2)$
- Assumption: Shared information for both causes:

Transfer to Competing Risks

- Setting with two outcomes/event types: $(\mathbf{t}_1, \delta_1), (\mathbf{t}_2, \delta_2)$
- Assumption: Shared information for both causes:
  - If X_j is important for cause 1, may also be relevant for cause 2

Transfer to Competing Risks

- Setting with two outcomes/event types: $(\mathbf{t}_1, \delta_1), (\mathbf{t}_2, \delta_2)$
- Assumption: Shared information for both causes:
  - If X_j is important for cause 1, may also be relevant for cause 2
  - $\Rightarrow$ lower its penalty in cause-specific models

Transfer to Competing Risks

- Setting with two outcomes/event types: $(\mathbf{t}_1, \delta_1), (\mathbf{t}_2, \delta_2)$
- Assumption: Shared information for both causes:
  - If X_j is important for cause 1, may also be relevant for cause 2
  - $\Rightarrow$ lower its penalty in cause-specific models
- Basic idea: Adapt previous algorithm to Cox regression

Transfer to Competing Risks

- Setting with two outcomes/event types: $(\mathbf{t}_1, \delta_1), (\mathbf{t}_2, \delta_2)$
- Assumption: Shared information for both causes:
  - If X_j is important for cause 1, may also be relevant for cause 2
  - $\Rightarrow$ lower its penalty in cause-specific models
- Basic idea: Adapt previous algorithm to Cox regression
- Multi-task $\simeq$ "Multi-cause"

Transfer to Competing Risks

- Setting with two outcomes/event types: $(\mathbf{t}_1, \delta_1), (\mathbf{t}_2, \delta_2)$
- Assumption: Shared information for both causes:
  - If X_j is important for cause 1, may also be relevant for cause 2
  - $\Rightarrow$ lower its penalty in cause-specific models
- Basic idea: Adapt previous algorithm to Cox regression
- Multi-task $\simeq$ "Multi-cause"

Transfer to Competing Risks

- Setting with two outcomes/event types: $(\mathbf{t}_1, \delta_1), (\mathbf{t}_2, \delta_2)$
- Assumption: Shared information for both causes:
  - If X_j is important for cause 1, may also be relevant for cause 2
  - $\Rightarrow$ lower its penalty in cause-specific models
- Basic idea: Adapt previous algorithm to Cox regression
- Multi-task $\simeq$ "Multi-cause"

Dubbed "Cooperative Penalized (Cox) Regression" (CooPeR)

High-Dimensional Variable Selection

- Simulation setup borrowed from Binder et al. (2009)

High-Dimensional Variable Selection

- Simulation setup borrowed from Binder et al. (2009)
- Context: Gene expression data with competing risk target

High-Dimensional Variable Selection

- Simulation setup borrowed from Binder et al. (2009)
- Context: Gene expression data with competing risk target
- $n = 400$, $p = 5000$, organized in 4 main blocks, few informative variables overall (16)

High-Dimensional Variable Selection

- Simulation setup borrowed from Binder et al. (2009)
- Context: Gene expression data with competing risk target
- n $=$ 400, p $=$ 5000, organized in 4 main blocks, few informative variables overall (16)
- Fit models, use $\mathbf{1}\{\hat{\beta}_j \neq 0\}$ as classification decision

High-Dimensional Variable Selection

- Simulation setup borrowed from Binder et al. (2009)
- Context: Gene expression data with competing risk target
- n $=$ 400, p $=$ 5000, organized in 4 main blocks, few informative variables overall (16)
- Fit models, use $\mathbf{1}\{\hat{\beta}_j \neq 0\}$ as classification decision
- Compare CooPeR, penalized Cox (glmnet), RSF (rsfrc), CoxBoost

Simulation of True Effects

- Block 1 (**Mutual**): 250 variables, $\rho \approx 0.5$ 4 vars w/ *same effect* (0.5) on both causes

Simulation of True Effects

- Block 1 (**Mutual**): 250 variables, $\rho \approx 0.5$ 4 vars w/ *same effect* (0.5) on both causes
- Block 2 (**Reversed**): 250 variables, $\rho \approx 0.35$ 4 w/ effect of 0.5 on cause 1 and -0.5 on cause 2

Simulation of True Effects

- Block 1 (**Mutual**): 250 variables, $\rho \approx 0.5$ 4 vars w/ *same effect* (0.5) on both causes
- Block 2 (**Reversed**): 250 variables, $\rho \approx 0.35$ 4 w/ effect of 0.5 on cause 1 and -0.5 on cause 2
- Block 3 (**Disjoint**): 500 variables, $\rho \approx 0.05$ 3.1: 4 w/ effect on *cause 1 only* 3.2: 4 w/ effect on *cause 2 only*

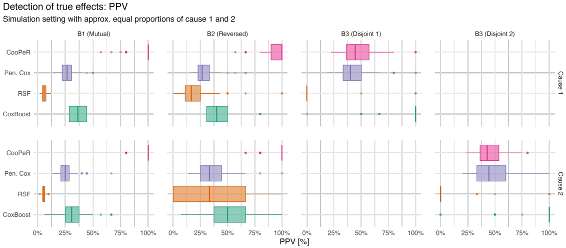
Simulation of True Effects

- Block 1 (**Mutual**): 250 variables, $\rho \approx 0.5$ 4 vars w/ *same effect* (0.5) on both causes
- Block 2 (**Reversed**): 250 variables, $\rho \approx 0.35$ 4 w/ effect of 0.5 on cause 1 and -0.5 on cause 2
- Block 3 (**Disjoint**): 500 variables, $\rho \approx 0.05$ 3.1: 4 w/ effect on *cause 1 only* 3.2: 4 w/ effect on *cause 2 only*
- Block 4 (**Cor. Noise**): 500 variables, $\rho \approx 0.32$

Simulation of True Effects

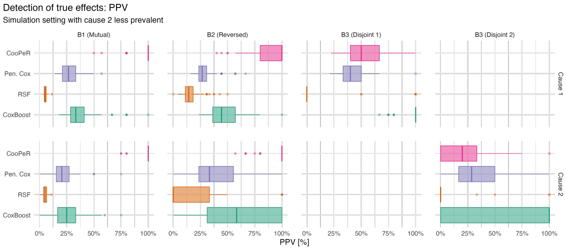
- Block 1 (**Mutual**): 250 variables, $\rho \approx 0.5$ 4 vars w/ *same effect* (0.5) on both causes
- Block 2 (**Reversed**): 250 variables, $\rho \approx 0.35$ 4 w/ effect of 0.5 on cause 1 and -0.5 on cause 2
- Block 3 (**Disjoint**): 500 variables, $\rho \approx 0.05$ 3.1: 4 w/ effect on *cause 1 only* 3.2: 4 w/ effect on *cause 2 only*
- Block 4 (**Cor. Noise**): 500 variables, $\rho \approx 0.32$
- Remaining variables: Uncorrelated noise

Positive Predictive Value



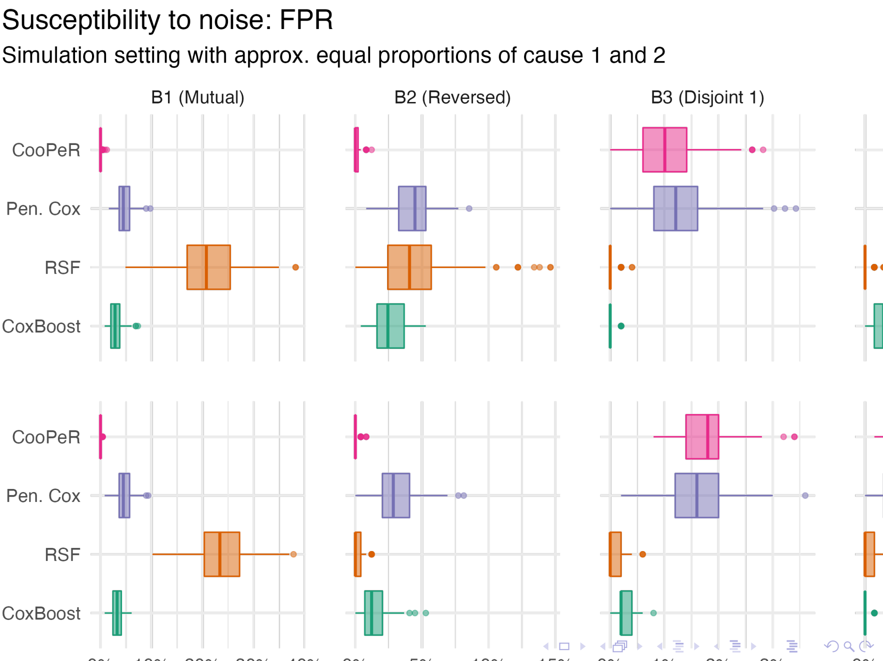
$$\mathrm{PPV} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

Positive Predictive Value $\text{PPV} = \frac{\text{TP}}{\text{TP+FP}}$

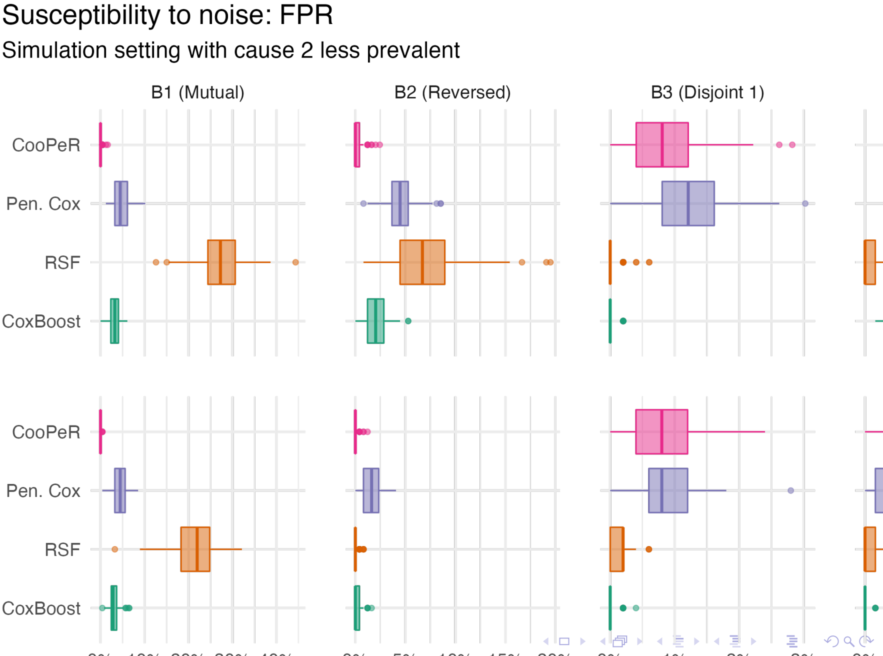


$$\text{PPV} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

False Positive Rate



False Positive Rate



How about real data?

- Still WIP

How about real data?

- Still WIP
- Idea:

How about real data?

- Still WIP
- Idea:
  1. Use algorithms for variable selection

How about real data?

- Still WIP
- Idea:
  1. Use algorithms for variable selection
  2. Fit standard cause-specific Cox model using only selected variables

How about real data?

- Still WIP
- Idea:
  1. Use algorithms for variable selection
  2. Fit standard cause-specific Cox model using only selected variables
  3. Evaluate prediction performance (tBrier, tAUC)

How about real data?

- Still WIP
- Idea:
  1. Use algorithms for variable selection
  2. Fit standard cause-specific Cox model using only selected variables
  3. Evaluate prediction performance (tBrier, tAUC)
- Tried bladder cancer data (Dyrskjøt et al. (2005)), did not go well

What's Next

- Would be nice to have a “working” real data example (Maybe TCGA)

What's Next

- Would be nice to have a "working" real data example (Maybe TCGA)
- Finishing the paper for journal submission

What's Next

- Would be nice to have a "working" real data example (Maybe TCGA)
- Finishing the paper for journal submission
- Talk about this at CEN in September

What's Next

- Would be nice to have a "working" real data example (Maybe TCGA)
- Finishing the paper for journal submission
- Talk about this at CEN in September

What's Next

- Would be nice to have a "working" real data example (Maybe TCGA)
- Finishing the paper for journal submission
- Talk about this at CEN in September

Open questions

- What does the actual optimization problem look like? (Does the algorithm converge? To what?)

What's Next

- Would be nice to have a "working" real data example (Maybe TCGA)
- Finishing the paper for journal submission
- Talk about this at CEN in September

Open questions

- What does the actual optimization problem look like? (Does the algorithm converge? To what?)
- What about $k > 2$ events? No trivial generalization

Thanks for listening!

References

Binder, Harald, Arthur Allignol, Martin Schumacher, and Jan Beyersmann. 2009. “Boosting for High-Dimensional Time-to-Event Data with Competing Risks.” *Bioinformatics* 25 (7): 890–96. https://doi.org/10.1093/bioinformatics/btp088.

Dyrskjøt, Lars, Karsten Zieger, Mogens Kruhøffer, Thomas Thykjaer, Jens L. Jensen, Hanne Primdahl, Natasha Aziz, Niels Marcussen, Klaus Møller, and Torben F. Orntoft. 2005. “A Molecular Signature in Superficial Bladder Carcinoma Predicts Clinical Outcome.” *Clinical Cancer Research: An Official Journal of the American Association for Cancer Research* 11 (11): 4029–36. https://doi.org/10.1158/1078-0432.CCR-04-2095.

Jacob, Laurent, Guillaume Obozinski, and Jean-Philippe Vert. 2009. “Group Lasso with Overlap and Graph Lasso.” In *Proceedings of the 26th Annual International Conference on Machine Learning - ICML ’09*, 1–8. Montreal, Quebec, Canada: ACM Press. https://doi.org/10.1145/1553374.1553431.

Tay, Jingyi Kenneth, Nima Aghaeepour, Trevor Hastie, and Robert Tibshirani. 2023. “Feature-Weighted Elastic Net: Using