BOOSTING IN COX REGRESSION

a comparison among the classical statistics-based, the likelihood-based and the model-based approaches

Created by Yue Xiong | Data Science

OUTLINE

- Introduction to the COX regression model
  - Background (Survival Analysis)
  - The COX model
  - FGD Boosting
- Application with the three approaches
  - Flexible boosting with the AFT model
  - mboost : model-based boosting in the COX model
  - CoxBoost : offset-based boosting in the COX model
- Comparison
- Allowing for mandatory variables
- Incorporating the pathway information
- Application with the mlr3 learner

INTRODUCTION TO THE COX REGRESSION MODEL (10MIN)

BACKGROUND: SURVIVAL ANALYSIS

- WHAT IS SURVIVAL ANALYSIS
- THE COX MODEL
- FGD BOOSTING

BACKGROUND: SURVIVAL ANALYSIS

WHAT IS SURVIVAL ANALYSIS

- statistics for analyzing the expected duration of time until one or more events of interest happen
  - e.g. death, failure in a mechanical system
- term
  - event
  - time: t
  - censoring observation: δ (missing data)
    - right-censoring
    - left-censoring

BACKGROUND: SURVIVAL ANALYSIS

THE COX MODEL

- intuition: for quantitive predictor variables, we use the COX PH hazard regression model
- hazard function λ
  - def: event rate at time t conditional or survival until time t or later
  - consider the time-to-event data (t, x, δ)
    - the hazard function $\lambda(t|X) = \lambda_0(t) exp(X^T \beta) = \lambda_0(t) exp(\beta_1 X_{i1} + ... + \beta_p X_{ip})$
- estimator for β
  - by maximizing the partial log-likelihood $(MPLE)$
  - $pl(\beta) = \sum_{i=1}^{n} \delta_i (X_i^T)\beta - log(\sum_{l \in R_i} exp\left\{(X_l^T \beta)\right\}$

BACKGROUND: SURVIVAL ANALYSIS

FGD BOOSTING

- <u>intuition</u>: at each iteration, a weak learner is fitted on the modified version of data with the goal of minimizing the empirical loss function
- <u>motivation</u>: we will use boosting techniques when
  - the number of covariates is large or
  - it is hard to directly derive the partial log-likelihood

BACKGROUND: SURVIVAL ANALYSIS

FGD BOOSTING

semi-parametric boosting

- given: $L(y, F(X))$ is a generic loss function and $F(x)$ is a statistical model.
- goal: to estimate $F(X)$ by iteratively updating its value through a base learner $h(y, X)$

BACKGROUND: SURVIVAL ANALYSIS

FGD BOOSTING

semi-parametric boosting

- **algorithm**
  1. initialize the estimate, e.g., $\hat{F}(X) = constant$;
  2. compute the pseudo-residual vector, $u = -\frac{\partial L(y, F(X))}{\partial F(X)}$, where $F(X) = \hat{F}(X)$;
  3. compute the update by:
     1. fit the base learner to the pseudo-residual vector, $\hat{h}(u, X)$;
     2. penalize the value, $\hat{f}(X) = v\hat{h}(u, X)$;
  4. update the estimate, $\hat{F}(X) = \hat{F}(X) + \hat{f}(X)$

BACKGROUND: SURVIVAL ANALYSIS

FGD BOOSTING

parametric boosting

- given: $F(X)$ is a parameterized class of functions, $F(X, \beta)$.
- the update process involves the estimate of the parameter, i.e. the regression coefficient β

BACKGROUND: SURVIVAL ANALYSIS

FGD BOOSTING

parametric boosting

- **algorithm**
  1. initialize the estimate, e.g., $\hat{\beta} = (0, \ldots, 0)$;
  2. compute the pseudo-residual vector, $u = -\frac{\partial L(y, F(X, \beta))}{\partial F(X, \beta)}$, where $\beta = \hat{\beta}$;
  3. compute the possible updates by:
     1. fit the base learner to the pseudo-residual vector, $\hat{h}(u, X_j)$;
     2. penalize the value, $\hat{b}_j = v\hat{h}(u, X_j)$;
  4. select the best update j^*
  5. update the estimate, $\hat{\beta}_{j^*} = \hat{\beta}_{j^*} + \hat{b}_j$

BACKGROUND: SURVIVAL ANALYSIS

FGD BOOSTING

boosting with the regularized empirical risk function

- penalize over the empirical risk function by adding $\lambda \cdot J(f)$
  - the complexity control parameter λ
  - the complexity penalty $J(f)$
- algorithm

??

APPLICATIONS WITH THE THREE APPROACHES (6MIN)

FLEXIBLE BOOSTING WITH THE AFT MODEL

FLEXIBLE BOOSTING WITH THE AFT MODEL

OBJECTIVE: TO FIT THE ACCELARATED FAILURE TIME MODEL

FLEXIBLE BOOSTING WITH THE AFT MODEL

OBJECTIVE: TO FIT THE ACCELARATED FAILURE TIME MODEL

$$log(T) = f(X) + \sigma \cdot w$$

FLEXIBLE BOOSTING WITH THE AFT MODEL

OBJECTIVE: TO FIT THE ACCELARATED FAILURE TIME MODEL

$$log(T) = f(X) + \sigma \cdot w$$

1. f is an unknown predictor function of $log(T)$,

FLEXIBLE BOOSTING WITH THE AFT MODEL

OBJECTIVE: TO FIT THE ACCELARATED FAILURE TIME MODEL

$$log(T) = f(X) + \sigma \cdot w$$

1. f is an unknown predictor function of $log(T)$, e.g. linear or additive function of covariates X;

FLEXIBLE BOOSTING WITH THE AFT MODEL

OBJECTIVE: TO FIT THE ACCELARATED FAILURE TIME MODEL

$$log(T) = f(X) + \sigma \cdot w$$

1. f is an unknown predictor function of $log(T)$, e.g. linear or additive function of covariates X;

2. σ is an unknown scale parameter

FLEXIBLE BOOSTING WITH THE AFT MODEL

OBJECTIVE: TO FIT THE ACCELARATED FAILURE TIME MODEL

$$log(T) = f(X) + \sigma \cdot w$$

1. f is an unknown predictor function of $log(T)$, e.g. linear or additive function of covariates X;

2. σ is an unknown scale parameter which controls the amount of noise added to f;

FLEXIBLE BOOSTING WITH THE AFT MODEL

OBJECTIVE: TO FIT THE ACCELARATED FAILURE TIME MODEL

$$log(T) = f(X) + \sigma \cdot w$$

1. f is an unknown predictor function of $log(T)$, e.g. linear or additive function of covariates X;

2. σ is an unknown scale parameter which controls the amount of noise added to f;

3. w is a noise variable

FLEXIBLE BOOSTING WITH THE AFT MODEL

OBJECTIVE: TO FIT THE ACCELARATED FAILURE TIME MODEL

$$log(T) = f(X) + \sigma \cdot w$$

1. f is an unknown predictor function of $log(T)$, e.g. linear or additive function of covariates X;

2. σ is an unknown scale parameter which controls the amount of noise added to f;

3. w is a noise variable whose distribution function determines the distribution form of survival time T.

FLEXIBLE BOOSTING WITH THE AFT MODEL

OBJECTIVE: TO FIT THE ACCELARATED FAILURE TIME MODEL

$$log(T) = f(X) + \sigma \cdot w$$

1. f is an unknown predictor function of $log(T)$, e.g. linear or additive function of covariates X;

2. σ is an unknown scale parameter which controls the amount of noise added to f;

3. w is a noise variable whose distribution function determines the distribution form of survival time T.

* If fully specified, we refer to it as a parametric AFT model.

FLEXIBLE BOOSTING WITH THE AFT MODEL

OBJECTIVE: TO FIT THE ACCELARATED FAILURE TIME MODEL

$$log(T) = f(X) + \sigma \cdot w$$

1. f is an unknown predictor function of $log(T)$, e.g. linear or additive function of covariates X;

2. σ is an unknown scale parameter which controls the amount of noise added to f;

3. w is a noise variable whose distribution function determines the distribution form of survival time T.

* If fully specified, we refer to it as a parametric AFT model.

* Otherwise, it is named a semi-parametric AFT model

FLEXIBLE BOOSTING WITH THE AFT MODEL

FLEXIBLE BOOSTING WITH THE AFT MODEL

FGD BOOSTING WITH THE SEMI-PARAMETRIC AFT MODEL

FLEXIBLE BOOSTING WITH THE AFT MODEL

FGD BOOSTING WITH THE SEMI-PARAMETRIC AFT MODEL

Similarly,

FLEXIBLE BOOSTING WITH THE AFT MODEL

FGD BOOSTING WITH THE SEMI-PARAMETRIC AFT MODEL

Similarly,

1. Initialize with f

FLEXIBLE BOOSTING WITH THE AFT MODEL

FGD BOOSTING WITH THE SEMI-PARAMETRIC AFT MODEL

Similarly,

1. Initialize with f
2. Compute the negative gradient vector predictor function of $log(T)$;

FLEXIBLE BOOSTING WITH THE AFT MODEL

FGD BOOSTING WITH THE SEMI-PARAMETRIC AFT MODEL

Similarly,

1. Initialize with f
2. Compute the negative gradient vector predictor function of $log(T)$;
3. Fit the negative gradient vector to each covariate of X seperately by p times

FLEXIBLE BOOSTING WITH THE AFT MODEL

FGD BOOSTING WITH THE SEMI-PARAMETRIC AFT MODEL

Similarly,

1. Initialize with f

2. Compute the negative gradient vector predictor function of $log(T)$;

3. Fit the negative gradient vector to each covariate of X seperately by p times using a simple linear regression estimator;

FLEXIBLE BOOSTING WITH THE AFT MODEL

FGD BOOSTING WITH THE SEMI-PARAMETRIC AFT MODEL

Similarly,

1. Initialize with f

2. Compute the negative gradient vector predictor function of $log(T)$;

3. Fit the negative gradient vector to each covariate of X seperately by p times using a simple linear regression estimator;

4. Select the best-fitted component of X,

FLEXIBLE BOOSTING WITH THE AFT MODEL

FGD BOOSTING WITH THE SEMI-PARAMETRIC AFT MODEL

Similarly,

1. Initialize with f
2. Compute the negative gradient vector predictor function of $log(T)$;
3. Fit the negative gradient vector to each covariate of X seperately by p times using a simple linear regression estimator;
4. Select the best-fitted component of X, according to a pre-specified goodness of fit criterion;

FLEXIBLE BOOSTING WITH THE AFT MODEL

FGD BOOSTING WITH THE SEMI-PARAMETRIC AFT MODEL

Similarly,

1. Initialize with f

2. Compute the negative gradient vector predictor function of $log(T)$;

3. Fit the negative gradient vector to each covariate of X seperately by p times using a simple linear regression estimator;

4. Select the best-fitted component of X, according to a pre-specified goodness of fit criterion;

5. Update with the real-valued step length factor v

FLEXIBLE BOOSTING WITH THE AFT MODEL

FGD BOOSTING WITH THE SEMI-PARAMETRIC AFT MODEL

Similarly,

1. Initialize with f

2. Compute the negative gradient vector predictor function of $log(T)$;

3. Fit the negative gradient vector to each covariate of X seperately by p times using a simple linear regression estimator;

4. Select the best-fitted component of X, according to a pre-specified goodness of fit criterion;

5. Update with the real-valued step length factor v and iterate m_{stop} times.

FLEXIBLE BOOSTING WITH THE AFT MODEL

FLEXIBLE BOOSTING WITH THE AFT MODEL

FGD BOOSTING WITH THE PARAMETRIC AFT MODEL

FLEXIBLE BOOSTING WITH THE AFT MODEL

FGD BOOSTING WITH THE PARAMETRIC AFT MODEL

Apart from f, we are also estimating for the scale parameter σ

FLEXIBLE BOOSTING WITH THE AFT MODEL

FGD BOOSTING WITH THE PARAMETRIC AFT MODEL

Apart from f, we are also estimating for the scale parameter σ

MBOOST FOR COX REGRESSION

MBOOST FOR COX REGRESSION

model-based approach where the loss function $L(\cdot)$ can be specified

MBOOST FOR COX REGRESSION

model-based approach where the loss function $L(\cdot)$ can be specified

suppose we are using the L2-loss function,

MBOOST FOR COX REGRESSION

model-based approach where the loss function $L(\cdot)$ can be specified

suppose we are using the L2-loss function, then the estimator for β is $(X^T X)^{-1} X^T u$

MBOOST FOR COX REGRESSION

model-based approach where the loss function $L(\cdot)$ can be specified

suppose we are using the L2-loss function, then the estimator for β is $(X^T X)^{-1} X^T u$

e.g. *glmboost*

MBOOST FOR COX REGRESSION

model-based approach where the loss function $L(\cdot)$ can be specified

suppose we are using the L2-loss function, then the estimator for β is $(X^T X)^{-1} X^T u$

e.g. *glmboost*

algorithm

MBOOST FOR COX REGRESSION

model-based approach where the loss function $L(\cdot)$ can be specified

suppose we are using the L2-loss function, then the estimator for β is $(X^T X)^{-1} X^T u$

e.g. $glmboost$

algorithm (initialize,

MBOOST FOR COX REGRESSION

model-based approach where the loss function $L(\cdot)$ can be specified

suppose we are using the L2-loss function, then the estimator for β is $(X^T X)^{-1} X^T u$

e.g. *glmboost*

algorithm (initialize, compute and fit,

MBOOST FOR COX REGRESSION

model-based approach where the loss function $L(\cdot)$ can be specified

suppose we are using the L2-loss function, then the estimator for β is $(X^T X)^{-1} X^T u$

e.g. *glmboost*

algorithm (initialize, compute and fit, select the best update,

MBOOST FOR COX REGRESSION

model-based approach where the loss function $L(\cdot)$ can be specified

suppose we are using the L2-loss function, then the estimator for β is $(X^T X)^{-1} X^T u$

e.g. $glmboost$

algorithm (initialize, compute and fit, select the best update, update the estimate)

COXBOOST FOR COX REGRESSION

COXBOOST FOR COX REGRESSION

offset-based approach and derive the penalized MPLE

COXBOOST FOR COX REGRESSION

offset-based approach and derive the penalized MPLE

in each boosting step, the previous step are incorporated as an offset in penalized log-likelihood estimation

COXBOOST FOR COX REGRESSION

offset-based approach and derive the penalized MPLE

in each boosting step, the previous step are incorporated as an offset in penalized log-likelihood estimation

COXBOOST FOR COX REGRESSION

offset-based approach and derive the penalized MPLE

in each boosting step, the previous step are incorporated as an offset in penalized log-likelihood estimation

e.g. add the L2-penalty

COXBOOST FOR COX REGRESSION

offset-based approach and derive the penalized MPLE

in each boosting step, the previous step are incorporated as an offset in penalized log-likelihood estimation

e.g. add the L2-penalty

$$pl_{pen}(\beta) = pl(\beta) + \lambda||\beta||_2^2 = pl(\beta) - 0.5\lambda\beta^T P\beta$$

COXBOOST FOR COX REGRESSION

offset-based approach and derive the penalized MPLE

in each boosting step, the previous step are incorporated as an offset in penalized log-likelihood estimation

e.g. add the L2-penalty

$$pl_{pen}(\beta) = pl(\beta) + \lambda ||\beta||_2^2 = pl(\beta) - 0.5\lambda\beta^T P\beta$$

an offset term $\hat{\eta} = X^T\hat{\beta}$ is added to the above log-likelihood to save the iterative updates of the parameter estimate

COXBOOST FOR COX REGRESSION

offset-based approach and derive the penalized MPLE

in each boosting step, the previous step are incorporated as an offset in penalized log-likelihood estimation

e.g. add the L2-penalty

$$pl_{pen}(\beta) = pl(\beta) + \lambda||\beta||_2^2 = pl(\beta) - 0.5\lambda\beta^T P\beta$$

an offset term $\hat{\eta} = X^T\hat{\beta}$ is added to the above log-likelihood to save the iterative updates of the parameter estimate

algorithm

COXBOOST FOR COX REGRESSION

offset-based approach and derive the penalized MPLE

in each boosting step, the previous step are incorporated as an offset in penalized log-likelihood estimation

e.g. add the L2-penalty

$$pl_{pen}(\beta) = pl(\beta) + \lambda||\beta||_2^2 = pl(\beta) - 0.5\lambda\beta^T P\beta$$

an offset term $\hat{\eta} = X^T\hat{\beta}$ is added to the above log-likelihood to save the iterative updates of the parameter estimate

algorithm (initialize,

COXBOOST FOR COX REGRESSION

offset-based approach and derive the penalized MPLE

in each boosting step, the previous step are incorporated as an offset in penalized log-likelihood estimation

e.g. add the L2-penalty

$$pl_{pen}(\beta) = pl(\beta) + \lambda ||\beta||_2^2 = pl(\beta) - 0.5\lambda\beta^T P\beta$$

an offset term $\hat{\eta} = X^T\hat{\beta}$ is added to the above log-likelihood to save the iterative updates of the parameter estimate

algorithm (initialize, compute updates with the penalized MPLE and fit,

COXBOOST FOR COX REGRESSION

offset-based approach and derive the penalized MPLE

in each boosting step, the previous step are incorporated as an offset in penalized log-likelihood estimation

e.g. add the L2-penalty

$$pl_{pen}(\beta) = pl(\beta) + \lambda||\beta||_2^2 = pl(\beta) - 0.5\lambda\beta^T P\beta$$

an offset term $\hat{\eta} = X^T\hat{\beta}$ is added to the above log-likelihood to save the iterative updates of the parameter estimate

algorithm (initialize, compute updates with the penalized MPLE and fit, select the best update,

COXBOOST FOR COX REGRESSION

offset-based approach and derive the penalized MPLE

in each boosting step, the previous step are incorporated as an offset in penalized log-likelihood estimation

e.g. add the L2-penalty

$$pl_{pen}(\beta) = pl(\beta) + \lambda ||\beta||_2^2 = pl(\beta) - 0.5\lambda\beta^T P\beta$$

an offset term $\hat{\eta} = X^T\hat{\beta}$ is added to the above log-likelihood to save the iterative updates of the parameter estimate

algorithm (initialize, compute updates with the penalized MPLE and fit, select the best update, update the estimate)

COMPARISON (5MIN)

FLEXIBLE BOOSTING WITH THE AFT MODEL

MBOOST

COXBOOST

FLEXIBLE BOOSTING WITH THE AFT MODEL

1. σ and w

MBOOST

COXBOOST

FLEXIBLE BOOSTING WITH THE AFT MODEL

1. σ and w

2. semi-parametric and parametric

MBOOST

COXBOOST

FLEXIBLE BOOSTING WITH THE AFT MODEL

1. σ and w

2. semi-parametric and parametric

MBOOST

1. user-defined loss function

COXBOOST

FLEXIBLE BOOSTING WITH THE AFT MODEL

1. σ and w

2. semi-parametric and parametric

MBOOST

1. user-defined loss function

2. MPLE

COXBOOST

FLEXIBLE BOOSTING WITH THE AFT MODEL

1. σ and w
2. semi-parametric and parametric

MBOOST

1. user-defined loss function
2. MPLE

COXBOOST

1. user-defined loss function

FLEXIBLE BOOSTING WITH THE AFT MODEL

1. σ and w

2. semi-parametric and parametric

MBOOST

1. user-defined loss function

2. MPLE

COXBOOST

1. user-defined loss function

2. restricted MPLE

FLEXIBLE BOOSTING WITH THE AFT MODEL

1. σ and w
2. semi-parametric and parametric

MBOOST

1. user-defined loss function
2. MPLE

COXBOOST

1. user-defined loss function
2. restricted MPLE
3. flexible penalty structure

FLEXIBLE BOOSTING WITH THE AFT MODEL

1. σ and w

2. semi-parametric and parametric

MBOOST

1. user-defined loss function

2. MPLE

COXBOOST

1. user-defined loss function

2. restricted MPLE

3. flexible penalty structure (variable selection,

FLEXIBLE BOOSTING WITH THE AFT MODEL

1. σ and w

2. semi-parametric and parametric

MBOOST

1. user-defined loss function

2. MPLE

COXBOOST

1. user-defined loss function

2. restricted MPLE

3. flexible penalty structure (variable selection, mandatory variables,

FLEXIBLE BOOSTING WITH THE AFT MODEL

1. σ and w

2. semi-parametric and parametric

MBOOST

1. user-defined loss function

2. MPLE

COXBOOST

1. user-defined loss function

2. restricted MPLE

3. flexible penalty structure (variable selection, mandatory variables, correlated covariates)

REMARK 1

REMARK 1

The three approaches are identical when

REMARK 1

The three approaches are identical when

REMARK 1

The three approaches are identical when

* f is set to be a linear regression model

REMARK 1

The three approaches are identical when

* f is set to be a linear regression model

* $L(\cdot)$ is defined as the same loss function (e.g. L2-Loss)

REMARK 1

The three approaches are identical when

* f is set to be a linear regression model

* $L(\cdot)$ is defined as the same loss function (e.g. L2-Loss)

* As for the AFT model: w is not specified

REMARK 1

The three approaches are identical when

* f is set to be a linear regression model

* $L(\cdot)$ is defined as the same loss function (e.g. L2-Loss)

* As for the AFT model: w is not specified

* As for the CoxBoost: λ is set to zero (no penalty is added)

REMARK 2 (NOT SURE)

REMARK 2 (NOT SURE)

The previous remark is also true for suitable values of v and λ

REMARK 2 (NOT SURE)

The previous remark is also true for suitable values of v and λ

REMARK 2 (NOT SURE)

The previous remark is also true for suitable values of v and λ

with standardized X

REMARK 2 (NOT SURE)

The previous remark is also true for suitable values of v and λ

with standardized X

* $\lambda = n(1 - v)/v$

REMARK 2 (NOT SURE)

The previous remark is also true for suitable values of v and λ

with standardized X

$$* \lambda = n(1 - v)/v$$

or

REMARK 2 (NOT SURE)

The previous remark is also true for suitable values of v and λ

with standardized X

- $\lambda = n(1 - v)/v$

or

- $\lambda = \frac{X_j^T X_j + vpl_{\beta_j}(0|\hat{\beta})}{v}$

REMARK 3

REMARK 3

The learning path of the three different approach-based boosting procedures may differ due to different choice of which dimension should be updated at each boosting step

REMARK 3

The learning path of the three different approach-based boosting procedures may differ due to different choice of which dimension should be updated at each boosting step

REMARK 3

The learning path of the three different approach-based boosting procedures may differ due to different choice of which dimension should be updated at each boosting step

in the AFT model

REMARK 3

The <u>learning path</u> of the three different approach-based boosting procedures may differ due to different choice of which dimension should be updated at each boosting step

in the AFT model

* extra paticipation of σ

REMARK 3

The learning path of the three different approach-based boosting procedures may differ due to different choice of which dimension should be updated at each boosting step

in the AFT model

* extra paticipation of σ

in $glmboost$

REMARK 3

The learning path of the three different approach-based boosting procedures may differ due to different choice of which dimension should be updated at each boosting step

in the AFT model

* extra paticipation of σ

in $glmboost$

* the choice is based on the residuals of the regression of u on X_j

REMARK 3

The learning path of the three different approach-based boosting procedures may differ due to different choice of which dimension should be updated at each boosting step

in the AFT model

* extra paticipation of σ

in glmboost

* the choice is based on the residuals of the regression of u on X_j

in CoxBoost

REMARK 3

The learning path of the three different approach-based boosting procedures may differ due to different choice of which dimension should be updated at each boosting step

in the AFT model

* extra paticipation of σ

in $glmboost$

* the choice is based on the residuals of the regression of u on X_j

in $CoxBoost$

* dimension is selected with the largest decrease of the penalized partial log-likelihood function

ALLOWING FOR MANDATORY VARIABLES (3MIN)

WHAT IS MANDATORY VARIABLES?

WHAT IS MANDATORY VARIABLES?

FAVORING STRATEGY IN THE *AFT MODEL*

WHAT IS MANDATORY VARIABLES?

FAVORING STRATEGY IN THE *AFT MODEL*

* LASSO regression approach

WHAT IS MANDATORY VARIABLES?

FAVORING STRATEGY IN THE *AFT MODEL*

* LASSO regression approach

FAVORING STRATEGY IN *glmboost*

WHAT IS MANDATORY VARIABLES?

FAVORING STRATEGY IN THE *AFT MODEL*

* LASSO regression approach

FAVORING STRATEGY IN *glmboost*

* The covariates are divided into the mandatory and the non-mandatory groups

WHAT IS MANDATORY VARIABLES?

FAVORING STRATEGY IN THE *AFT MODEL*

* LASSO regression approach

FAVORING STRATEGY IN *glmboost*

* The covariates are divided into the mandatory and the non-mandatory groups

* the penalization with v is applied only to the non-mandatory components

WHAT IS MANDATORY VARIABLES?

FAVORING STRATEGY IN THE *AFT MODEL*

* LASSO regression approach

FAVORING STRATEGY IN $glmboost$

* The covariates are divided into the mandatory and the non-mandatory groups

* the penalization with v is applied only to the non-mandatory components

FAVORING STRATEGY IN $CoxBoost$

WHAT IS MANDATORY VARIABLES?

FAVORING STRATEGY IN THE *AFT MODEL*

* LASSO regression approach

FAVORING STRATEGY IN $glmboost$

* The covariates are divided into the mandatory and the non-mandatory groups

* the penalization with v is applied only to the non-mandatory components

FAVORING STRATEGY IN $CoxBoost$

1. configure the penalty matrix P

WHAT IS MANDATORY VARIABLES?

FAVORING STRATEGY IN THE *AFT MODEL*

* LASSO regression approach

FAVORING STRATEGY IN $glmboost$

* The covariates are divided into the mandatory and the non-mandatory groups

* the penalization with v is applied only to the non-mandatory components

FAVORING STRATEGY IN $CoxBoost$

1. configure the penalty matrix P

2. mandatory variables can be introduced by updating their parameters before each step of componentwise $CoxBoost$

INCORPORATING PATHWAY INFORMATION (1MIN)

WHAT IS PATHWAY INFORMATION?

WHAT IS PATHWAY INFORMATION?

* correlated microarray features

WHAT IS PATHWAY INFORMATION?

* correlated microarray features

* trade-off between model complexity and the representation power

WHAT IS PATHWAY INFORMATION?

* correlated microarray features

* trade-off between model complexity and the representation power

OBJECTIVE: DISCOURAGING THE SELECTION OF SINGLE MICROARRAY FEATURES

WHAT IS PATHWAY INFORMATION?

* correlated microarray features

* trade-off between model complexity and the representation power

OBJECTIVE: DISCOURAGING THE SELECTION OF SINGLE MICROARRAY FEATURES

1. increasing the penalty for a selected covariate

WHAT IS PATHWAY INFORMATION?

* correlated microarray features

* trade-off between model complexity and the representation power

OBJECTIVE: DISCOURAGING THE SELECTION OF SINGLE MICROARRAY FEATURES

1. increasing the penalty for a selected covariate

2. decreasing the penalty for connected covariates

APPLICATION WITH THE MLR3 LEARNER

a^2