Package ‘MST’

February 19, 2015

Type Package

Title Multivariate Survival Trees

Version 1.0

Date 2014-09-19

Author Xiaogang Su, Peter Calhoun, and Juanjuan Fan

Maintainer Peter Calhoun <calhoun.peter@gmail.com>

Description Constructs trees for multivariate survival data using marginal and frailty models

License GPL-2

Imports survival, MASS

NeedsCompilation no

Repository CRAN

Date/Publication 2014-09-20 07:14:13

R topics documented:

MST-package .......................................................... 2
bootstrap.grow.prune ................................................ 3
bootstrap.size .......................................................... 4
grow.MST .............................................................. 5
MST ................................................................. 6
MST.plot .............................................................. 10
prune.size.testsample ................................................ 11
rmultime ............................................................... 12

Index 14
Description

This package constructs trees for multivariate survival data using marginal and frailty models.

Details

- **Package:** MST
- **Type:** Package
- **Version:** 1.0
- **Date:** 2014-09-19
- **License:** GPL-2

Regression trees require few statistical assumptions, handle a variety of data structures, and provide meaningful interpretations. There are several functions to construct regression trees (see `rpart`, `tree`, and `party`); this package extends the implementation to multivariate survival data. There are two main approaches to analyzing correlated failure times. One is the marginal approach studied by authors Wei et al. (1989) and Liang et al. (1993). In the marginal model, the correlation is modeled implicitly using estimating equations on the marginal distribution formulated by the Cox (1972) proportional hazards model. The other approach is the frailty model studied by Clayton (1978) and Clayton and Cuzick (1985). In the frailty model, the correlation is modeled explicitly by a multiplicative random effect called frailty, which corresponds to some common unobserved characteristics shared by all correlated times.

Construction of the tree consists of three stages: growing the initial tree, pruning the tree, and selecting the best-sized subtree; details of these steps are described elsewhere (Fan et al. [2006], Su and Fan [2004], and Fan et al. [2009]). There are two methods for selecting the best-sized subtree. When the dataset is large, one may divide the dataset into a training sample to grow and prune the initial tree and a test sample to select the best-sized tree. When the dataset is small, one can resample the dataset to choose the best-sized subtree.

Author(s)

Xiaogang Su, Peter Calhoun, & Juanjuan Fan

Maintainer: Peter Calhoun <calhoun.peter@gmail.com>

References


---

**bootstrap.grow.prune**  
*Bootstrap Approach to Grow and Prune Multivariate Survival Trees*

**Description**

Grows and prunes multivariate survival trees using a bootstrap approach

**Usage**

```r
bootstrap.grow.prune(data, method=c("marginal", "gamma.frailty", "exp.frailty"), 
min.ndsz=20, n0=3, col.time, col.status, col.id,col.split.var, col.ctg=NULL, 
max.depth=10, mtry=length(col.split.var), B=30, LeBlanc=TRUE, min.boot.tree.size=1, 
details=FALSE, cont.split=c("distinct","percentiles"), delta=0.05, nCutPoints=50)
```

**Arguments**

- `B`: Number of bootstrap samples
- `data`: Dataset
- `method`: Indicates method of estimating correlation: must be either "marginal", "gamma.frailty", or "exp.frailty"
- `min.ndsz`: Number: Controls the minimum node size
- `n0`: Number: Controls the minimum number of uncensored event times at either child node
- `col.time`: Column number for time
- `col.status`: Column number for status
- `col.id`: Column number for id
col.split.var Column numbers of all variables wanting to split
col.ctg Column numbers of categorical variables wanting to split; should be a subset of col.split.var
max.depth Number: Maximum depth of tree
mtry Number of variables considered at each split
LeBlanc Logical: Indicates if entire sample used (alternative is out-of-bag sample)
min.boot.tree.size Number: Minimum size of tree grown at each bootstrap
details Logical: Indicates if detailed information should be printed
cont.split Indicates candidate splits: must be "distinct" or "percentiles"
delta Consider cutpoints from delta to 1 − delta. Only used when cont.split="percentiles"
nCutPoints Number of cutpoints (percentiles) considered. Only used when cont.split="percentiles"

Value

initial.tree The initial tree
boot.prune Information on trees pruned for each bootstrap sample
boot.tree Pruned trees for each bootstrap sample

Note
This function is computationally intensive and may take some time

Author(s)
Xiaogang Su, Peter Calhoun, Juanjuan Fan

See Also
MST

Description
Selects the best-sized multivariate survival tree using a bootstrap approach

Usage

bootstrap.size(bootstrap.grow.prune.results, plot.it = TRUE, filename = NULL, horizontal = TRUE)
Arguments

- bootstrap.grow.prune.results: Results from bootstrap.grow.prune.
- plot.it: Logical: Indicates if you want to plot the goodness-of-fit vs. tree size.
- filename: Name of the file plotted.
- horizontal: Logical: Indicates if plot should be landscape.

Value

- initial.tree: The initial tree.
- btree: The best-sized tree structure based on the penalty used.

Author(s)

Xiaogang Su, Peter Calhoun, Juanjuan Fan

See Also

- MST

---

**Description**

Grows the initial multivariate survival tree.

**Usage**

```r
grow.MST(dat, test = NULL, method = c("marginal", "gamma.frailty", "exp.frailty"),
min.ndsz = 20, n0 = 3, col.time, col.status, col.id, col.split.var, col.ctg = NULL,
max.depth = 10, mtry = length(col.split.var), details = FALSE,
cont.split = c("distinct", "percentiles"), delta=0.05, nCutPoints=50)
```

**Arguments**

- dat: Data or training sample.
- test: Test sample if available.
- method: Indicates method of estimating correlation: must be either "marginal", "gamma.frailty", or "exp.frailty".
- min.ndsz: Number: Controls the minimum node size.
- n0: Number: Controls the minimum number of uncensored event times at either child node.
- col.time: Column number for time.
MST

col.status  Column number for status
col.id  Column number for id
col.split.var  Column numbers of all variables wanting to split
col.ctg  Column numbers of categorical variables wanting to split; should be a subset of col.split.var
max.depth  Number: Maximum depth of tree
mtry  Number of variables considered at each split
details  Logical: Indicates if detailed information should be printed
cont.split  Indicates candidate splits: must be "distinct" or "percentiles"
delta  Consider cutpoints from delta to 1 − delta. Only used when cont.split="percentiles"
nCutPoints  Number of cutpoints (percentiles) considered. Only used when cont.split="percentiles"

Value

This function outputs the initial tree

Author(s)

Xiaogang Su, Peter Calhoun, Juanjuan Fan

See Also

MST

Description

Constructs trees for multivariate survival data using marginal and frailty models. Combines grow.MST, prune.size.testsample, bootstrap.grow.prune, and bootstrap.size into one final wrapper function.

Usage

MST(training, test = NULL, method = c("marginal", "gamma.frailty", "exp.frailty"), min.ndsz = 20, n0 = 3, col.time, col.status, col.id, col.split.var, col.ctg = NULL, max.depth = 10, mtry = length(col.split.var), selection.method = c("test.sample", "bootstrap"), B = 30, LeBlanc = TRUE, min.boot.tree.size = 1, plot.it = TRUE, filename = NULL, horizontal = TRUE, details = FALSE, cont.split = c("distinct", "percentiles"), delta=0.05, nCutPoints=50)
**Arguments**

- **training**: Data or training sample
- **test**: Test sample if available
- **method**: Indicates method of estimating correlation: must be either "marginal", "gamma.frailty", or "exp.frailty"
- **min.ndsz**: Number: Controls the minimum node size
- **n0**: Number: Controls the minimum number of uncensored event times at either child node
- **col.time**: Column number for time
- **col.status**: Column number for status
- **col.id**: Column number for id
- **col.split.var**: Column numbers of all variables wanting to split
- **col.ctg**: Column numbers of categorical variables wanting to split; should be a subset of col.split.var
- **max.depth**: Number: Maximum depth of tree
- **mtry**: Number of variables considered at each split
- **selection.method**: Indicates method of selecting the best-sized subtree: "test.sample" or "bootstrap"
- **B**: Number of bootstrap samples. Only used if selection.method="bootstrap"
- **LeBlanc**: Logical: Indicates if entire sample used (alternative is out-of-bag sample). Only used if selection.method="bootstrap"
- **min.boot.tree.size**: Number: Minimum size of tree grown at each bootstrap
- **plot.it**: Logical: Indicates if you want to plot the goodness-of-fit vs. tree size
- **filename**: Name of the file plotted
- **horizontal**: Logical: Indicates if plot should be landscape
- **details**: Logical: Indicates if detailed information should be printed
- **cont.split**: Indicates candidate splits: must be "distinct" or "percentiles"
- **delta**: Consider cutpoints from delta to 1 − delta. Only used when cont.split="percentiles"
- **nCutPoints**: Number of cutpoints (percentiles) considered. Only used when cont.split="percentiles"

**Details**

Marginal and frailty models are the two main ways to analyze correlated failure times. Let \( Z_{ij} \) represent the covariate vector for the \( j \)th member in the \( i \)th cluster.

The marginal model uses the Cox (1972) proportional hazards model:

\[
\lambda_{ij}(t|Z_{ij}) = \lambda_0(t) \exp(\beta \cdot I(Z_{ij} \leq c))
\]

where \( \lambda_0(t) \) is an unspecified baseline hazard function and \( I(\cdot) \) is the indicator function.
The gamma frailty model uses the proportional hazards model:

$$\lambda_{ij}(t|Z_{ij}, w_i) = \lambda_0(t) \exp(\beta \cdot I(Z_{ij} \leq c))w_i$$

where $\lambda_0(t)$ is an unspecified baseline hazard function, $I(\cdot)$ is the indicator function, and $w_i$ is the frailty term for the $i$th cluster.

The exponential frailty model uses the proportional hazards model:

$$\lambda_{ij}(t|Z_{ij}, w_i) = \exp(\beta_0 + \beta_1 \cdot I(Z_{ij} \leq c))w_i$$

where $I(\cdot)$ is the indicator function and $w_i$ is the frailty term for the $i$th cluster.

For the marginal model, a robust logrank statistic is calculated for each covariate $Z$ and possible cutpoint $c$. The estimate of the score function and likelihood of $\beta$ can be obtained assuming independence. However, the variance-covariance structure adjusts for the dependence using a sandwich-type estimator. The best split is the one with the largest robust logrank statistic.

For the frailty models, a score test statistic is calculated from the maximum integrated log likelihood for each covariate $Z$ and possible cutpoint $c$. The frailty term must follow some known positive distribution; one common choice is $w_i \sim \Gamma(1/\nu, 1/\nu)$ where $\nu$ represents an unknown variance. Note, the exponential frailty model replaces the baseline hazard function with a constant, yielding different score test statistics and typically computationally faster splits. The best split is the one with the largest score test statistic.

Note, for continuous variables with many distinct cutpoints, the number of cutpoints considered can be reduced to percentiles. Using percentiles increases efficiency at the expense of less accuracy.

Growing the initial tree is done by splitting nodes (as described above) reiteratively until the maximum depth of the tree is reached or a small number of observations remain at terminal node. However, as the final tree model can be any subtree of the initial tree, the number of subtrees can become massive. A goodness-of-fit with an added penalty for the number of internal nodes is used to prune the trees (i.e. reduce the number of subtrees considered). The best-sized tree is selected by the largest goodness-of-fit with the added penalty using either the test sample or bootstrap samples.

Value

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tree0</td>
<td>The initial tree</td>
</tr>
<tr>
<td>pruning.info</td>
<td>Trees pruned and considered in the best tree selection</td>
</tr>
<tr>
<td>best.tree.size</td>
<td>The best tree size based on the penalty used</td>
</tr>
<tr>
<td>best.tree.structure</td>
<td>The best tree structure based on the penalty used</td>
</tr>
</tbody>
</table>

Note, $G$ represents the goodness-of-fit statistic with the respective penalties. Nodes are represented using binary numbers. Observations with $vname \leq$ cut moves to the left of the tree; otherwise moves to the right. See MST.plot

Warning

Error messages in the gamma.frailty model sometimes occur when using the bootstrap method. Increasing min ndsz may help fix these errors.
Note

Code may take awhile to implement large datasets. To decrease computation time, user should use test sample (selection.method="test.sample"). User can also split continuous variables based on percentiles (cont.split="percentiles") at the expense of slightly less accuracy. Method="gamma.frailty" is more computationally intensive.

Author(s)

Xiaogang Su, Peter Calhoun, and Juanjuan Fan

References


See Also

MST.plot, rpart

Examples

```r
set.seed(409871)
# Training dataset:
training <- rmultime(beta=c(-1, 2, 2, 0, 0), cutoff=c(.5, .5, 0, 0), digits=1, icensor=1, model = "marginal.multivariate.exponential", rho=.65, N=200, K=4)$dat

# Test dataset:
test <- rmultime(beta=c(-1, 2, 2, 0, 0), cutoff=c(.5, .5, 0, 0), digits=1, icensor=1, model = "marginal.multivariate.exponential", rho=.65, N=100, K=4)$dat

# Construct Multivariate Survival Tree:
fit <- MST(training=training, test=test, method="marginal", min.ndsz=30, n0=5, col.time=3, col.status=4, col.id=1, col.split.var=5:8, col.ctg=col.ctg, max.depth=6, mtry=length(col.split.var), selection.method = "test.sample", plot.it=TRUE, filename=NULL, horizontal=TRUE, details=FALSE, cont.split="distinct")
```
MST.plot  

Plot Multivariate Survival Trees

Description

Plot trees from the MST output

Usage

MST.plot(tree, textDepth = 3, lines = "rectangle")

Arguments

tree  Tree to be plotted

textDepth  Depth of text

lines  Shape of lines connecting nodes: must be either "rectangle" or "triangle"

Value

Plot of tree structure

Note

This function gives a simple, efficient method of plotting the multivariate survival trees. True statements in the plot move to the left. This function does not provide much flexibility when customizing the trees. User may need to format figures manually

Author(s)

Peter Calhoun, Xiaogang Su, Juanjuan Fan

See Also

tree, rpart

Examples

set.seed(409871)
#Training dataset:
training <- rmultime(beta=c(-1, 2, 2, 0, 0), cutoff=c(.5, .5, 0, 0),
digits=1, icensor=1, model = "marginal.marginal.multivariate.exponential",
rho=.65, N=200, K=4)$dat

#Test dataset:
test <- rmultime(beta=c(-1, 2, 2, 0, 0), cutoff=c(.5, .5, 0, 0),
digits=1, icensor=1, model = "marginal.marginal.multivariate.exponential",
rho=.65, N=100, K=4)$dat
# Construct Multivariate Survival Tree:
fit <- MST(training=training, test=test, method="marginal", min.ndsz=30, n0=5,
col.time=3, col.status=4, col.id=1, col.split.var=5:8, col.ctg=col.ctg,
max.depth=6, mtry=length(col.split.var), selection.method = "test.sample",
plot.it=TRUE, filename=NULL, horizontal=TRUE, details=FALSE, cont.split="distinct")

MST.plot(fit$tree0, textDepth=5)  # Plot of initial tree
MST.plot(fit$best.tree.structure[[4]])  # Plot of best-sized tree

---

**prune.size.testsample**  *Prunes and Selects Best-Sized Multivariate Survival Tree*

**Description**

Prunes the initial tree and selects the best-sized multivariate survival tree

**Usage**

```r
prune.size.testsample(tree)
```

**Arguments**

- `tree`  
  The initial tree

**Value**

- `result`  
  Information on trees pruned
- `btree`  
  The best-sized tree structures

**Author(s)**

Xiaogang Su, Peter Calhoun, Juanjuan Fan

**See Also**

MST
**Random Multivariate Survival Data**

**Description**
Generates multivariate survival data

**Usage**
```
rmultime(beta = c(-1, 2, 1, 0, 0), cutoff = c(0.5, 0.5, 0, 0), digits = 1, icensor = 1, model = c("gamma.frailty", "log.normal.frailty", "marginal.multivariate.exponential", "marginal.nonabsolutely.continuous", "nonPH.weibull"), v = 1, rho = 0.65, a = 1.5, lambda = 0.1, N = 100, K = 4)
```

**Arguments**
- **beta**: Vector of beta coefficients (first number is baseline hazard coefficient \( \beta_0 \), remaining numbers are slope coefficients for covariates \( \beta_1 \))
- **cutoff**: Cutoff values for each covariate
- **digits**: Rounding digits
- **icensor**: Control for censoring rate: 1 - 50%
- **model**: Model for simulating data: must be either "gamma.frailty", "log.normal.frailty", "marginal.multivariate.exponential", "marginal.nonabsolutely.continuous", or "nonPH.weibull"
- **v**: Scale parameter for "gamma.frailty" and "nonPH.weibull" or variance parameter for "log.normal.frailty" models. Not used in marginal models
- **rho**: Correlation for marginal models. Not used in other models
- **a**: Parameter for "nonPH.weibull" model. Not used in other models
- **lambda**: Parameter for "nonPH.weibull" model. Not used in other models
- **N**: Number of clusters (ids)
- **K**: Number of units per cluster

**Details**
This function generates multivariate survival data. Letting \( i = 1, \ldots, N \) number of clusters and \( j = 1, \ldots, K \) number of units per cluster, the following multivariate survival models can be used:

- **gamma.frailty**: \( \lambda_{ij}(t) = \exp(\beta_0 + \beta_1 \cdot I(Z_{ij} \leq c)) w_i \) with \( w_i \sim \Gamma(1/v, 1/v) \)
- **log.normal.frailty**: \( \lambda_{ij}(t) = \exp(\beta_0 + \beta_1 \cdot I(Z_{ij} \leq c) + w_i) \) with \( w_i \sim N(0, v) \)
- **marginal.multivariate.exponential**: \( \lambda_{ij}(t) = \exp(\beta_0 + \beta_1 \cdot I(Z_{ij} \leq c)) \) absolutely continuous
- **marginal.nonabsolutely.continuous**: \( \lambda_{ij}(t) = \exp(\beta_0 + \beta_1 \cdot I(Z_{ij} \leq c)) \) not absolutely continuous
nonPH.weibull: \[ \lambda_{ij}(t) = \lambda_0(t) \exp(\beta_0 + \beta_1 \cdot I(Z_{ij} \leq c)) w_i \] with \[ w_i \sim \Gamma(1/v, 1/v) \] and \[ \lambda_0(t) = \alpha t^{\alpha-1} \]

The user specifies the coefficients (\( \beta_0 \) and \( \beta_1 \)), the cutoff values, the censoring rate, and the model with the respective parameters.

Value
- dat: The simulated data
- model: The model used

Author(s)
- Xiaogang Su, Peter Calhoun, Juanjuan Fan

References

See Also
- genSurv, complex.surv.dat.sim, survsim

Examples
```r
randMarginalExp<-rmultime(beta=c(-1, 2, 0, 0), cutoff=c(.5, .5, 0, 0),
digits=1, icensor=1, model = "marginal.multivariate.exponential",
rho=.65, N=200, K=4)$dat

randFrailtyGamma<-rmultime(beta=c(-1, 1, 3), cutoff=c(.4, .6, 0, 0),
digits=1, icensor=1, model = "gamma.frailty",
v=1, N=200, K=4)$dat
```
Index

*Topic **Bootstrap**
bootstrap.grow.prune, 3
bootstrap.size, 4

*Topic **Correlated**
bootstrap.grow.prune, 3
bootstrap.size, 4
grow.MST, 5
MST, 6
MST.plot, 10
prune.size.testsamp, 11
rmultime, 12

*Topic **Multivariate**
bootstrap.grow.prune, 3
bootstrap.size, 4
grow.MST, 5
MST, 6
MST.plot, 10
prune.size.testsamp, 11
rmultime, 12

*Topic **Plot**
MST.plot, 10

*Topic **Simulation**
rmultime, 12

*Topic **Survival**
bootstrap.grow.prune, 3
bootstrap.size, 4
grow.MST, 5
MST, 6
MST.plot, 10
prune.size.testsamp, 11
rmultime, 12

*Topic **Trees**
bootstrap.grow.prune, 3
bootstrap.size, 4
grow.MST, 5
MST, 6
MST.plot, 10
prune.size.testsamp, 11

as.numeric.factor (MST-package), 2
bootstrap.grow.prune, 3, 6
bootstrap.size, 4, 6
de (MST-package), 2
grow (MST-package), 2
grow.MST, 5, 6
is.odd (MST-package), 2
loglik0 (MST-package), 2
MST, 4–6, 6, 11
MST-package, 2
MST.plot, 8, 9, 10
obtain.btree (MST-package), 2
ordinalize (MST-package), 2
partition.MST (MST-package), 2
power.set (MST-package), 2
prune.size (MST-package), 2
prune.size.testsamp, 6, 11
rmultime, 12
send.down (MST-package), 2
splitting.stat.MST1 (MST-package), 2
splitting.stat.MST2 (MST-package), 2