Package ‘MST’

February 16, 2018

Type Package
Title Multivariate Survival Trees
Version 2.1
Author Xiaogang Su [aut],
Peter Calhoun [aut, cre],
Juanjuan Fan [aut]
Maintainer Peter Calhoun <calhoun.peter@gmail.com>
Description Constructs trees for multivariate survival data using marginal and frailty models.
      Grows, prunes, and selects the best-sized tree.
License GPL-2
Depends R (>= 3.1.0), survival
Imports graphics, grDevices, MASS, Formula, methods, partykit, stats
LazyData true
NeedsCompilation no
Repository CRAN
Date/Publication 2018-02-16 19:29:08 UTC

R topics documented:

MST-package .................................................. 2
getTree ...................................................... 3
MST .......................................................... 4
rmultime ..................................................... 7
Teeth ......................................................... 9

Index 12
Description

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

Details

Package: MST
Type: Package
Version: 2.1
Date: 2018-02-16
License: GPL-2

Decision trees require few statistical assumptions, handle a variety of data structures, and provide meaningful interpretations. There are several R packages that provide functions to construct survival trees (see \texttt{rpart}, \texttt{partykit}, and \texttt{DStree}); 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 generalized 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.

The construction of the tree adopts a modified CART procedure controlling for the correlated failure times. The procedure 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


### getTree

**Extract initial or best-sized tree**

#### Description

This function extracts the tree based on the split penalty.

#### Usage

```r
getTree(mstObj, Ga = c("0", "2", "3", "4", "log_n"))
```

#### Arguments

- `mstObj`: The output from the MST fit
- `Ga`: The split penalty

#### Value

The tree of object class "constparty"

#### Author(s)

Peter Calhoun <calhoun.peter@gmail.com>

#### See Also

- `MST`
Description

Constructs trees for multivariate survival data using marginal and frailty models. A wrapper function that grows a large initial tree, prunes the tree, and selects the best sized tree.

Usage

\[
\text{MST(formula, data, test = NULL, weights_data, weights_test, subset, }
\text{ method = c("marginal", "gamma.frailty", "exp.frailty", "stratified", "independence"),}
\text{ msplit = 20, minevents = 3, minbucket = round(msplit/3), maxdepth = 10,}
\text{ mtry = NULL, distinct = TRUE, delta = 0.05, nCutPoints = 50,}
\text{ selection.method = c("test.sample", "bootstrap"),}
\text{ B = 30, LeBlanc = TRUE, min.boot.tree.size = 1,}
\text{ plot.Ga = TRUE, filename = NULL, horizontal = TRUE, details = FALSE, sortTrees = TRUE)}
\]

Arguments

formula A linear survival model with the response on the left of a ~ operator and the predictors, separated by + operators, on the right. Cluster (or id) variable is distinguished by a vertical bar | (e.g. Surv(time, status) ~ x1 + x2 | id). Categorical predictors must be treated as a factor.
data Data to grow and prune the tree
test Test sample if available
weights_data An optional vector of weights to grow the tree
weights_test An optional vector of weights to select the best-sized tree
subset An optional vector specifying a subset of observations to be used to grow the tree
method Indicates method of handling correlation: must be either "marginal", "gamma.frailty", "exp.frailty", "stratified", or "independence"
msplit Number: Controls the minimum node size
minevents Number: Controls the minimum number of uncensored event times
minbucket Number: Controls the minimum number of observations in any terminal node
maxdepth Number: Maximum depth of tree
mtry Number of variables considered at each split. The default is to consider all variables
distinct Logical: Indicates if all distinct cutpoints or only percentiles considered
delta Consider cutpoints from delta to 1 - delta. Only used when distinct = TRUE
nCutPoints Number of cutpoints (percentiles) considered. Only used when distinct = TRUE
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.Ga
Logical: Indicates if goodness-of-fit vs. tree size should be plotted
filename
Name of the file plotted
horizontal
Logical: Indicates if plot should be landscape
details
Logical: Indicates if detailed information on the construction should be printed
sortTrees
Logical: Indicates if trees should be sorted such that each split to the left has
lower risk of failure

Details

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

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

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

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:

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

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:

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

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 \( X \) 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 \( X \) 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.

Stratified model grows a tree by minimizing the within-strata variation. This method should be used with care because the tree will not split on variables with a fixed value within each stratum. The independence model ignores the dependence and uses the logrank statistic as the splitting rule.
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

- tree0: The initial tree. Tree listed as constparty object
- pruning.info: Trees pruned and considered in the best tree selection
- best.tree.size: The best tree size based on the penalty used
- best.tree.structure: The best tree structure based on the penalty used. Tree listed as constparty object

Note, the constparty object requires a constant fit from each terminal node. Thus, the predict and plot functions ignore the dependence, so users are recommended to fit their own model when making predictions (see example).

Warning

Error messages in the gamma frailty models sometimes occur when using the bootstrap method. Increasing minsplit may help fix these errors. The exponential frailty model can have problems for large, extremely unbalanced designs. Currently weights can only be applied to marginal and gamma frailty models.

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 (distinct = FALSE) at the expense of slightly less accuracy. Gamma frailty models are more computationally intensive.

Author(s)

Xiaogang Su, Peter Calhoun, and Juanjuan Fan

References


See Also

rpart

Examples

```r
set.seed(186117)
data <- rmultime(N = 200, K = 4, beta = c(-1, 0.8, 0.8, 0, 0), cutoff = c(0.5, 0.3, 0, 0),
model = "marginal.multivariate.exponential", rho = 0.65)$dat
test <- rmultime(N = 100, K = 4, beta = c(-1, 0.8, 0.8, 0, 0), cutoff = c(0.5, 0.3, 0, 0),
model = "marginal.multivariate.exponential", rho = 0.65)$dat

#Construct Multivariate Survival Tree:
fit <- mst(formula = Surv(time, status) ~ x1 + x2 + x3 + x4 | id, data, test,
method = "marginal", msplit = 100, minevents = 20, selection.method = "test.sample")

(tree_final <- getTree(fit, 4))
plot(tree_final)

#Fit a model from the final tree
data$term_nodes <- as.factor(predict(tree_final, newdata = data, type = 'node'))
coxph(Surv(time, status) ~ term_nodes + cluster(id), data = data)
```

---

**rmultime**  
*Random Multivariate Survival Data*

**Description**

Generates multivariate survival data

**Usage**

```r
rmultime(N = 100, K = 4, 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)
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>Number of clusters (ids)</td>
</tr>
<tr>
<td>K</td>
<td>Number of units per cluster</td>
</tr>
<tr>
<td>beta</td>
<td>Vector of beta coefficients (first number is baseline hazard coefficient ($\beta_0$), remaining numbers are slope coefficients for covariates ($\beta_1$))</td>
</tr>
</tbody>
</table>
### Details

This function generates multivariate survival data. Letting \( i = 1, \ldots, N \) number of clusters, \( j = 1, \ldots, K \) number of units per cluster, and \( X_{ij} \) be a candidate covariate, the following multivariate survival models can be used:

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

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
c randMarginalExp <- rmultinom(N = 200, K = 4, beta = c(-1, 2, 0, 0), cutoff = c(0.5, 0.5, 0, 0), digits = 1, icensor = 1, model = "multivariate.exponential", rho = .65)$dat
c randFrailtyGamma <- rmultinom(N = 200, K = 4, beta = c(-1, 1, 3, 0), cutoff = c(0.4, 0.6, 0), digits = 1, icensor = 1, model = "frailtygamma", v = 1)$dat
d
```

Teeth

Tooth Loss Data

Description

Survival of teeth with various predictors.

Usage

```r
data("Teeth")
```

Format

A data frame with 65,890 teeth on the following 56 variables.

- `x1` numeric. `mobil` Mobility score (on a scale 0–5).
- `x2` numeric. `bleed` Bleeding on Probing (percentage).
- `x3` numeric. `plaque` Plaque Score (percentage).
- `x4` numeric. `pocket_mean` Periodontal Probing Depth (tooth-level mean).
- `x5` numeric. `pocket_max` Periodontal Probing Depth (tooth-level max).
- `x6` numeric. `cal_mean` Clinical Attachment Level (tooth-level mean).
- `x7` numeric. `cal_max` Clinical Attachment Level (tooth-level max).
- `x8` numeric. `fgm_mean` Free Gingival Margin (tooth-level mean).
- `x9` numeric. `fgm_max` Free Gingival Margin (tooth-level max).
- `x10` numeric. `mg` Mucogingival Defect.
- `x11` numeric. `filled` Filled Surfaces.
- `x12` numeric. `decay_new` Decayed Surfaces – new.
- `x13` numeric. `decay_recur` Decayed Surfaces – recurrent.
- `x14` numeric. `dfs` Decayed and Filled Surfaces.
- `x15` factor. `crown` Crown.
- `x16` factor. `endo` Endodontic Therapy.
- `x17` factor. `implant` Tooth Implant.
- `x18` factor. `pontic` Bridge Pontic.
- `x19` factor. `missing_tooth` Missing Tooth.
- `x20` factor. `filled_tooth` Filled Tooth.
x21 factor. decayed_tooth Decayed Tooth.
x22 factor. furc_max Furcation Involvement for Molars.
x23 numeric. bleed_ave Bleeding on Probing (mean percentage).
x24 numeric. plaque_ave Plaque Index (mean percentage).
x25 numeric. pocket_mean_ave Periodontal Probing Depth (mean of tooth mean).
x26 numeric. pocket_max_ave Periodontal Probing Depth (mean of tooth max).
x27 numeric. cal_mean_ave Clinical Attachment Level (mean of tooth mean).
x28 numeric. cal_max_ave Clinical Attachment Level (mean of tooth max).
x29 numeric. fgm_mean_ave Free Gingival Margin (mean of tooth max).
x30 numeric. fgm_max_ave Free Gingival Margin (mean of tooth max).
x31 numeric. mg_ave Mucogingival Defect (mean).
x32 numeric. filled_sum Filled Surfaces (total).
x33 numeric. filled_ave Filled Surfaces (mean).
x34 numeric. decay_new_sum New Decayed Surfaces (total).
x35 numeric. decay_new_ave New Decayed Surfaces (mean).
x36 numeric. decay_recur_sum Recurrent Decayed Surfaces (total).
x37 numeric. decay_recur_ave Recurrent Decayed Surfaces (mean).
x38 numeric. dfs_sum Decayed and Filled Surfaces (total).
x39 numeric. dfs_ave Decayed and Filled Surfaces (mean).
x40 numeric. filled_tooth_sum Number of Filled Teeth.
x41 numeric. filled_tooth_ave Percentage of Filled Teeth.
x42 numeric. decayed_tooth_sum Number of Decayed Teeth.
x43 numeric. decayed_tooth_ave Percentage of Decayed Teeth.
x44 numeric. missing_tooth_sum Number of Missing Teeth.
x45 numeric. missing_tooth_ave Percentage of Missing Teeth.
x46 numeric. total_tooth Number of Teeth.
x47 numeric. dft Number of Decayed and Filled Teeth.
x48 numeric. baseline_age Patient Age at Baseline (years).
x49 factor. gender Gender.
x50 factor. diabetes Diabetes Mellitus.
x51 factor. tobacco_ever Tobacco Use.
molar logical. Molar.
id numeric. Patient ID.
tooth numeric. Tooth ID.
event numeric. Tooth Loss Status.
time numeric. Follow Up Time.
Details

Patients were treated at the Creighton University School of Dentistry from August 2007 to March 2013. This is a subset of the original data.

The goal is to estimate the survival time of teeth (molars or non-molars) using 51 predictors (22 tooth-level factors (x1–x22) and 29 patient-level factors (x23–x51)).

Examples

data(Tooths)
Index

*Topic **Correlated**
  MST, 4
  rmultime, 7
*Topic **Multivariate**
  MST, 4
  rmultime, 7
*Topic **Simulation**
  rmultime, 7
*Topic **Survival**
  MST, 4
  rmultime, 7
*Topic **Trees**
  MST, 4
*Topic **datasets**
  Teeth, 9

as.numeric.factor (MST-package), 2
bootstrap.grow.prune (MST-package), 2
bootstrap.size (MST-package), 2
de (MST-package), 2

getTree, 3
grow0 (MST-package), 2
grow.MST (MST-package), 2
is.odd (MST-package), 2
listIntoParty (MST-package), 2
listIntoTree (MST-package), 2
loglik0 (MST-package), 2

MST, 3, 4
MST-package, 2
MST.plot (MST-package), 2

obtain.btree (MST-package), 2
ordinalizeFunc (MST-package), 2
partition.MST (MST-package), 2

power.set (MST-package), 2
prune.size (MST-package), 2

rmultime, 7
send.down (MST-package), 2
sortTree (MST-package), 2
splitting.stat.MST1 (MST), 4
splitting.stat.MST2 (MST), 4
splitting.stat.MST3 (MST), 4
splitting.stat.MST4 (MST), 4

Teeth, 9