Package: jlctree (via r-universe)

October 22, 2024

Title Joint Latent Class Trees for Joint Modeling of Time-to-Event and Longitudinal Data

Date 2021-04-12

Version 0.0.2

Author Ningshan Zhang and Jeffrey S. Simonoff

Maintainer Ningshan Zhang <nzhang@stern.nyu.edu>

Depends survival, rpart, lme4

Encoding UTF-8

License GPL

Description Implements the tree-based approach to joint modeling of time-to-event and longitudinal data. This approach looks for a tree-based partitioning such that within each estimated latent class defined by a terminal node, the time-to-event and longitudinal responses display a lack of association. See Zhang and Simonoff (2018) <arXiv:1812.01774>.

RoxygenNote 6.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2021-04-15 11:10:02 UTC

Contents

`jlctree-package` *Fits Joint Latent Class Tree (JLCT) model.*

Description

Fits Joint Latent Class Tree (JLCT) model. The main function of this package is `jlctree`.

Problem setup: The dataset contains three types of variables about each subject: the time-to-event, the longitudinal outcome, and additional covariates. The goal is to jointly model the time-to-event by a survival model and the longitudinal outcomes by a linear mixed-effects model, and using the additional covariates. The longitudinal outcomes consist of repeated measurements, thus are expected to be time-varying for a given subject. The additional covariates can be either time-invariant or time-varying. Nevertheless, `jlctree` also allows data with time-invariant longitudinal outcome and covariates.

JLCT model: This package implements the Joint Latent Class Tree (JLCT) modeling approach. JLCT assumes that the population consists of homogeneous latent classes; within a latent class subjects follow the same survival and linear mixed-effects model, but those differ from class to class. In addition, JLCT assumes that conditioning on latent class membership, time-to-event and longitudinal outcomes are independent. JLCT looks for a tree-based partitioning such that within each estimated latent class defined by a terminal node, the time-to-event and longitudinal responses display a lack of association. Once the tree is constructed, JLCT assigns each observation to a latent class (i.e. terminal node), and independently fits survival and linear mixed-effects models, using the class membership information.

Time-to-event data format: The time-to-event data format required by `jlctree` depends on the time-varying nature of the variables to use: if longitudinal outcome, or any of the covariates specified in `survival`, `classmb`, `fixef`, and `ranef` is time-varying, then the time-to-event data must be in left-truncated right-censored (LTRC) format. Otherwise, when longitudinal outcome and all of the covariates are time-invariant, there should be only one observation per subject, and the time-to-event data can either be in LTRC format (when there exits subject-specific entry time) or in standard right-censored format.

To construct time-to-event data in left-truncated right-censored format, consider using function `tmerge` in `R` package `survival`. See the simulated `data_timevar` and `data_timeinv` for examples of LTRC format and right-censored format respectively.

References

Ningshan Zhang and Jeffrey S. Simonoff: Joint Latent Class Trees: A Tree-Based Approach to Joint Modeling of Time-to-event and Longitudinal Data. arXiv:1812.01774 (2018).

See Also

`jlctree`, `data_timeinv`, `data_timevar`

`data_timeinv` *A simulated dataset with time-invariant longitudinal outcome and covariates.*

Description

A simulated dataset with time-invariant longitudinal outcome, time-to-event, and time-invariant covariates. Since longitudinal outcome and all of the covariates are time-invariant, there is only one observation per subject. The time-to-event data is right-censored.

Usage

```
data(data_timeinv)
```

Format

A data frame with 500 rows and 10 variables.

ID subject identifier (1 - 500)

X1 continuous covariate between 0 and 1; time-invariant

X2 continuous covariate between 0 and 1; time-invariant

X3 binary covariate; time-invariant

X4 continuous covariate between 0 and 1; time-invariant

X5 categorical covariate taking values from 1, 2, 3, 4, 5; time-invariant

time_Y right-censored event time

delta censoring indicator, 1 if censored and 0 otherwise

y longitudinal outcome; time-invariant

g true latent class identifier 1, 2, 3, 4, which is determined by the outcomes of $1\{X1 > 0.5\}$ and $1\{X2 > 0.5\}$, with some noise

Examples

```
# The data for the first five subjects (ID = 1 - 5):
#
#  ID   X1   X2 X3  X4 X5    time_Y delta         y g
#   1 0.27 0.53  1 0.8  1 10.703940     0 0.8923776 2
#   2 0.37 0.68  1 0.5  3  9.153915     1 0.6871529 2
#   3 0.57 0.38  1 0.2  1  4.489658     1 0.8410745 3
#   4 0.91 0.95  0 0.4  3  1.009941     1 2.1058681 4
#   5 0.20 0.12  0 0.8  5 11.125094     0 0.1383508 1
```

data_timevar *A simulated dataset with time-varying longitudinal outcome and covariates.*

Description

A simulated dataset with time-varying longitudinal outcome, time-to-event, and time-varying covariates. The dataset is already converted into left-truncated right-censored (LTRC) format, so that the Cox model with time-varying longitudinal outcome as a covariate can be fit. See, for example, Fu and Simonoff (2017).

Usage

```
data(data_timevar)
```

Format

A data frame with 866 rows and 11 variables. The variables are as follows:

ID subject identifier (1 - 500)

X1 continuous covariate between 0 and 1; time-varying

X2 continuous covariate between 0 and 1; time-varying

X3 binary covariate; time-varying

X4 continuous covariate between 0 and 1; time-varying

X5 categorical covariate taking values from 1, 2, 3, 4, 5; time-varying

time_L left-truncated time

time_Y right-censored time

delta censoring indicator, 1 if censored and 0 otherwise

y longitudinal outcome; time-varying

g true latent class identifier 1, 2, 3, 4, which is determined by the outcomes of $1\{X1 > 0.5\}$ and $1\{X2 > 0.5\}$, with some noise

References

Fu, W. and Simonoff, J. S. (2017). Survival trees for left-truncated and right-censored data, with application to time-varying covariate data. Biostatistics, 18(2), 352-369.

Examples

```
# The data for the first five subjects (ID = 1 - 5):
#
#  ID   X1   X2 X3  X4 X5     time_L   time_Y delta          y g
#   1 0.27 0.53  0 0.0  4 0.09251632 1.536030     0 -0.2191137 1
#   1 0.49 0.71  1 0.0  5 1.53603028 4.366769     1  0.6429496 2
#   2 0.37 0.68  1 0.4  4 0.44674406 1.203560     0  0.5473454 2
#   2 0.65 0.67  0 0.2  5 1.20355968 1.330767     1  1.5515773 4
#   3 0.57 0.38  0 0.2  4 0.82944637 1.267248     0  1.1410397 3
#   3 0.79 0.19  1 0.4  4 1.26724819 5.749602     1  1.0888787 3
#   4 0.91 0.95  0 0.9  1 0.81237396 1.807741     1  2.2105303 4
#   5 0.20 0.12  1 0.3  5 0.80510669 1.029981     0 -0.1167814 1
#   5 0.02 0.31  0 0.4  5 1.02998145 6.404183     1 -0.1747389 1
```

get_lrt *Computes the likelihood ratio test statistic.*

Description

Computes the likelihood ratio test statistic. Not to be called directly by the user.

Usage

```
get_lrt(f1, f2, data, stable = TRUE, cov.max = 1e+05)
```

Arguments

`f1`	a two-sided formula of the fitted survival model, without the longitudinal outcome in the right side of the formula.
`f2`	a two-sided formula of the fitted survival model, same as `f1` but with the longitudinal outcome being the first covariate on the right side of the formula.
`data`	a data.frame containing the covariates in both `f1` and `f2`.
`stable`	a parameter, see also `jlctree.control`.
`cov.max`	a parameter, see also `jlctree.control`.

Value

The likelihood ratio test statistic.

See Also

`get_node_val`

Examples

```
data(data_timevar);
f1 <- Surv(time_L, time_Y, delta)~X3+X4+X5;
f2 <- Surv(time_L, time_Y, delta)~y+X3+X4+X5;
get_lrt(f1, f2, data_timevar);
```

get_node_val *Computes the test statistic at the current node.*

Description

Computes the test statistic at the current node. Not to be called directly by the user.

Usage

```
get_node_val(f1, f2, data, lrt = TRUE, ...)
```

Arguments

f1	a two-sided formula of the fitted survival model, without the longitudinal outcome in the right side of the formula. Only needed when lrt=TRUE.
f2	a two-sided formula of the fitted survival model, same as f1 but with the longitudinal outcome being the first covariate on the right side of the formula.
data	a data.frame containing covariates in f2.
lrt	if TRUE, use likelihood ratio test, otherwise use Wald test. Default is TRUE.
...	further arguments to pass to or from other methods.

Value

The test statistic at the current node.

See Also

get_lrt,get_wald

Examples

```
data(data_timevar);
f1 <- Surv(time_L, time_Y, delta)~X3+X4+X5;
f2 <- Surv(time_L, time_Y, delta)~y+X3+X4+X5;
get_node_val(f1, f2, data_timevar, lrt=TRUE);
```

get_wald *Computes the Wald test statistic.*

Description

Computes the Wald test statistic. Not to be called directly by the user.

Usage

```
get_wald(f, data)
```

Arguments

f	a two-sided formula of the fitted survival model, with the longitudinal outcome being the first covariate on the right side of the formula.
data	a data.frame containing covariates in f.

Value

The Wald test statistic.

See Also

get_node_val

Examples

```
data(data_timevar);
f <- Surv(time_L, time_Y, delta)~y+X3+X4+X5;
get_wald(f, data_timevar);
```

jlctree *Fits Joint Latent Class Tree (JLCT) model.*

Description

Fits Joint Latent Class Tree model. This is the main function that is normally called by the user. See jlctree-package for more details.

Usage

```
jlctree(survival, classmb, fixed, random, subject, data, parms = list(),
  control = list())
```

Arguments

`survival` a two-sided formula object; required. The left side of the formula corresponds to a `Surv()` object of type “counting” for left-truncated right-censored (LTRC) data, or of type “right” for right-censored data. The right side of the formula specifies the names of covariates to include in the survival model, excluding the longitudinal outcome.

`classmb` one-sided formula describing the covariates in the class-membership tree construction; required. Covariates used for tree construction are separated by + on the right of ~.

`fixed` two-sided linear formula object for the fixed-effects in the linear mixed-effects model for longitudinal outcomes; required. The longitudinal outcome is on the left of ~ and the covariates are separated by + on the right of ~.

`random` one-sided formula for the node-specific random effects in the linear mixed-effects model for longitudinal outcomes; optional. If missing, there are no node-specific random effects in the fitted linear mixed-effects model. Covariates with a random effect are separated by + on the right of ~.

`subject` name of the covariate representing the subject identifier; optional. If missing, there are no subject-specific random intercepts in the fitted linear mixed-effects model for longitudinal outcomes.

`data` the dataset; required.

`parms` parameter list of Joint Latent Class Tree model parameters. See also `jlctree.control`.

`control` `rpart` control parameters. See also `rpart.control`.

Value

A list with components:

`tree` an `rpart` object, containing the constructed Joint Latent Class tree.

`control` the `rpart.control` parameters.

`parms` the `jlctree.control` parameters.

`lmmmodel` an `lme4` object, containing the linear mixed-effects effects model with fixed effects, node-specific random effects (if valid), and subject-specific random intercepts (if valid). Returned when `fity` is TRUE.

`coxphmodel_diffh_diffs`
a `coxph` object, containing a Cox PH model with different hazards and different slopes across terminal nodes. Returned when `fits` is TRUE.

`coxphmodel_diffh`
a `coxph` object, containing a Cox PH model with different hazards but same slopes across terminal nodes. Returned when `fits` is TRUE.

`coxphmodel_diffs`
a `coxph` object, containing a Cox PH model with same hazards but different slopes across terminal nodes. Returned when `fits` is TRUE.

See Also

`jlctree-package`, `jlctree.control`, `rpart.control`

Examples

```
# Time-to-event in LTRC format:
data(data_timevar)
tree <- jlctree(survival=Surv(time_L, time_Y, delta)~X3+X4+X5,
                classmb=~X1+X2, fixed=y~X1+X2+X3+X4+X5, random=~1,
                subject='ID',data=subset(data_timevar, ID<=30),
                parms=list(maxng=4, fity=FALSE, fits=FALSE))

# Time-to-event in right-censored format:
data(data_timeinv)
tree <- jlctree(survival=Surv(time_Y, delta)~X3+X4+X5,
                classmb=~X1+X2, fixed=y~X1+X2+X3+X4+X5, random=~1,
                subject='ID', data=subset(data_timeinv, ID<=30),
                parms=list(maxng=4, fity=FALSE, fits=FALSE))
```

`jlctree.control` *Sets the control parameters for* `jlctree`.

Description

Sets the control parameters for `jlctree`.

Usage

```
jlctree.control(test.stat = "lrt", stop.thre = 3.84, stable = TRUE,
  maxng = 6, min.nevents = 5, split.add = 20, cov.max = 1e+05,
  fity = TRUE, fits = TRUE, ...)
```

Arguments

`test.stat` test statistic to use, “lrt” for likelihood ratio test, and “wald” for Wald test. Default is “lrt”.

`stop.thre` stops splitting if current node has test statistic less than `stop.thre`. Default is 3.84.

`stable` if TRUE, check the variance of the estimated coefficients in survival models fit at tree nodes. If a node has variance larger than `cov.max`, the splitting function will not consider splits leading to that node. Default is TRUE.

`maxng` maximum number of terminal nodes. Default is 6.

`min.nevents` minimum number of events in any terminal node. By default, this parameter is set to the number of covariates used in the survival model.

`split.add` when computing the difference between parent node’s test statistic and sum of child nodes’ test statistics, add `split.add` to the difference. When `split.add` > 0, tree may still split even if current split leads to negative improvement. Set `split.add` to a large positive value for the purpose of greedy splitting. Default is 20.

`cov.max` upper bound on the variance of the estimated coefficients in survival models at tree nodes. Default is 1e5.

`fity` if TRUE, once a tree is constructed, fit a linear mixed-effects model using tree nodes as group indicators. Default is TRUE.

`fits` if TRUE, once a tree is constructed, fit survival models using tree nodes as group indicators. Default is TRUE.

`...` further arguments to pass to or from other methods.

Value

A list of all these parameters.

See Also

`jlctree`,`jlctree-package`

`prune_tree` *Prunes an* `rpart` *tree to have the desired number of nodes.*

Description

Prunes an `rpart` tree to have the desired number of nodes.

Usage

```
prune_tree(tree, maxn)
```

Arguments

`tree` the tree to prune, an `rpart` object.

`maxn` desired number of terminal nodes.

Value

The pruned tree, an `rpart` object.

surve *Defines the evaluation function for a new splitting method of* `rpart`.

Description

Defines the evaluation function for a new splitting method of `rpart`. Not to be called directly by the user.

Usage

```
surve(y, wt, parms)
```

Arguments

`y`	the response value as found in the formula that is passed in by `rpart`. Note that `rpart` will normally have removed any observations with a missing response.
`wt`	the weight vector from the call, if any.
`parms`	the vector or list (if any) supplied by the user as a `parms` argument to the call.

Value

See reference.

References

https://cran.r-project.org/package=rpart/vignettes/usercode.pdf

See Also

`survs`,`survi`

survi *Defines the initialization function for a new splitting method of* `rpart`.

Description

Defines the initialization function for a new splitting method of `rpart`. Not to be called directly by the user.

Usage

```
survi(y, offset, parms, wt)
```

Arguments

`y` the response value as found in the formula that is passed in by `rpart`. Note that `rpart` will normally have removed any observations with a missing response.

`offset` the offset term, if any, found on the right hand side of the formula that is passed in by `rpart`.

`parms` the vector or list (if any) supplied by the user as a `parms` argument to the call.

`wt` the weight vector from the call, if any.

Value

See reference.

References

https://cran.r-project.org/package=rpart/vignettes/usercode.pdf

See Also

`survs`,`surve`

`survs` *Defines the splitting function for a new splitting method of* `rpart`.

Description

Defines the splitting function for a new splitting method of `rpart`. Not to be called directly by the user.

Usage

```
survs(y, wt, x, parms, continuous)
```

Arguments

`y` the response value as found in the formula that is passed in by `rpart`. Note that `rpart` will normally have removed any observations with a missing response.

`wt` the weight vector from the call, if any.

`x` vector of `x` values.

`parms` the vector or list (if any) supplied by the user as a `parms` argument to the call.

`continuous` if TRUE the `x` variable should be treated as continuous. The value of this parameter is determined by `rpart` automatically.

Value

See reference.

References

https://cran.r-project.org/package=rpart/vignettes/usercode.pdf

See Also

surve, survi

Index