Package ‘kaps’

February 20, 2015

Title K-Adaptive Partitioning for Survival data

Date 2014-11-01

Version 1.0.2

Description This package provides some routines to conduct the K-adaptive parititioning (kaps) algorithm for survival data. A function kaps is an implementation version of our algorithm.

Depends R (>= 3.0.0), methods, graphics, survival, Formula, coin

Suggests locfit

LazyData yes

License GPL-3

Maintainer Soo-Heang Eo <eo.sooheang@gmail.com>

NeedsCompilation yes

Author Soo-Heang Eo [aut, cre],
Seung-Mo Hong [ctb],
HyungJun Cho [aut]

Repository CRAN

Date/Publication 2014-11-01 15:20:40

R topics documented:

kaps-package 2
count.mindat 2
kaps 3
kaps-class 6
kaps.control 7
kapsNews 8
km.curve 9
plot 9
predict 10
print 11
show 11
summary 12
toy 12

Index **13**

kaps-package *K-adaptive partitioning for survival data.*

Description

This package provides some routines to conduct a K-adaptive partitioning algorithm, which divides the dataset into K heterogeneous subgroups based on the information from a prognostic factor.

Details

Package:	kaps
Type:	Package
Version:	1.0.2
Date:	2014-11-01
License:	GPL-3
LazyLoad:	no

This package contains some routines to conduct a *K*-adaptive parititioning for survival data (kaps) algorithm. A function `kaps()` is an implementation version of our algorithm which provides minimax-based partitioing rule.

Author(s)

Soo-Heang Eo <hanansh@korea.ac.kr>
Seung-Mo Hong <smhong28@gmail.com>
HyungJun Cho <hj4cho@korea.ac.kr>

References

S-H Eo, S-M Hong and H Cho (2014). *K*-adaptive partitioning for survival data, *submitted*.

See Also

`kaps`

count.mindat *Caculate the minimum sample size when the number of subgroups is given*

Description

This function caculates the minimum sample size of each partition when the number of subgroups is given.

Usage

```
count.mindat(formula, data, part = 10)
```

Arguments

`formula`	a Formula object with a response on the left hand side of the '~' operator, and the covariate terms on the right side. The response has to be a survival object with survival time and censoring status in the Surv function. For more details, see Formula page.
`data`	a data frame with variables used in formula. It needs at least three variables including survival time, censoring status, and a covariate. Multivariate covariates can be supported with "+" sign.
`part`	a numeric object to determine the number of subgroups we want to split.

See Also

`kaps`

Examples

```
data(toy)
count.mindat(Surv(time,staus) ~ meta, data = toy, part =5)
count.mindat(Surv(time,staus) ~ meta, data = toy, part =10)
count.mindat(Surv(time,staus) ~ meta, data = toy)
```

`kaps` *K-adaptive partitioing for survival data*

Description

Conduct K-adaptive partitioning algorithm for survival data

Usage

```
kaps(formula, data, K = 2:4, mindat, type = c("perm", "NULL"), ...)
```

Arguments

`formula` Formula object with a response on the left hand side of the '~' operator, and the covariate terms on the right side. The response has to be a survival object with survival time and censoring status in the Surv function. For more details, see Formula page.

`data` data frame with variables used in formula. It needs at least three variables including survival time, censoring status, and a covariate. Multivariate covariates can be supported with "+" sign.

`K` number of subgroups used in the model fitting. The default value is 2:4 which means finding optimal subgroups ranging from 2 to 4.

`type` Select a type of algorithm in order to find optimal number of subgroups. Two options are provided: `perm` and `NULL`. The `perm` chooses subgroups using permutation procudures, while the `NULL` passes a optimal selection algorithm.

`mindat` the minimum number of observations at each subgroup. The default value is 5% of observations.

`...` a list of tuning parameters with the class, "kapsOptions". For more details, see kaps.control.

Details

This function provides routines to conduct KAPS algorithm which is designed to classify cut-off values by the minimax-based rule.

Value

The function returns an object with class "kaps" with the following slots.

`call:` evaluated function call

`formula:` formula to be used in the model fitting

`data:` data to be used in the model fitting

`groupID:` information about the subgroup classification

`index:` an index for the optimal subgroup among the candidate K

`X:` test statistic with the worst pair of subgroups for the split set s

`Z:` the overall test staitstic with K subgroups using the split set s

`pair:` selected pair of subgroups

`split.var:` selected covariate in the model fitting

`split.pt:` selected set of cut-off points

`mindat:` minimum number of observations at a subgroup

`test.stat:` Bonferroni corrected p-value matrix. The first row means overall p-values and the second one denotes p-values of the worst-pair against K. The column in the matrix describes the order of K.

`over.stat.sample:`
adjusted overall test statistic by Bootstrapping

`pair.stat.sample:` adjusted worst-pair test statistic by Bootstrapping

`groups:` candidate K used in the argument

`results:` list object about the results of each candidate K

`Options:` tuning parameters

Author(s)

Soo-Heang Eo <eo.sooheang@gmail.com>
Seung-Mo Hong <smhong28@gmail.com>
HyungJun Cho <hj4cho@korea.ac.kr>

References

S-H Eo, S-M Hong and H Cho (2014). K-adaptive partitioning for survival data, *submitted*.

See Also

`show`, `plot`, `predict`, `print` and `summary` for the convenient use of kaps()
`kaps.control` to control kaps() more detail
`count.mindat` to calculate minimum subgroup sample size

Examples

```
## Not run:
  data(toy)
  f <- Surv(time, status) ~ meta
  # Fit kaps algorithm without cross-validation.
  # It means the step to finding optimal K is not entered.
  fit1 <- kaps(f, data = toy, K = 3)

  # show the object of kaps (it contains apss S4 class)
  fit1

  # plot Kaplan-Meire estimates
  plot(fit1)

  # Fit kaps algorithm for selection optimal number of subgropus.
  fit2 <- kaps(f, data = toy, K= 2:4)
  fit2

  # plot outputs with subgroup selection
  require(locfit) # for scatterplot smoothing
  plot(fit2)

  print(fit2,K=2)
  summary(fit2)
  summary(fit2,K=2)

  # require(party)
```

```
    # fit4 <- ctree(f, data = toy)

## End(Not run)
```

---

kaps-class *Class* "kaps"

Description

A S4 class for *K*-adaptive partitioning for survival data (kaps).

Objects from the Class

Objects can be created by calls of the form `new("kaps")`. The most important slot is `groupID`, which is a vector consisting of the information about classified subgroups.

Slots

`call`**:** evaluated function call

`formula`**:** formula used in the model fitting

`data`**:** data used in the model fitting

`groupID`**:** information about the classified subgroup

`index`**:** index for the optimal subgroup among the candidate K

`X`**:** test statistic with the worst pair of subgroups for the split set

`Z`**:** overall test statistic with K subgroups using the split set

`pair`**:** selected pair of subgroups

`split.var`**:** selected covariate in the model fitting

`split.pt`**:** selected set of cut-off points

`mindat`**:** minimum number of observations at a subgroup

`test.stat`**:** Bonferroni corrected p-value matrix. The first row means overall p-values and the second one denotes p-values of the worst-pair against K. The column in the matrix describes the order of K.

`over.stat.sample`**:** adjusted overall test statistic by Bootstrapping

`pair.stat.sample`**:** adjusted worst-pair test statistic by Bootstrapping

`groups`**:** candidate K used in the argument

`results`**:** a list of results about each K

`Options`**:** tuning parameters

Methods

- **show** `signature(object = "kaps")`: Same as the show method without the optional arguments
- **print** `signature(x = "kaps", K)`: Same as the print method with the specified number of subgroups K.
- **plot** `signature(x = "kaps", K)`: Plot an object
- **predict** `signature(object = "kaps")`: Predict an object by the estimated cut-off points
- **summary** `signature(object = "kaps")`: Summarize an object by survival times for each subgroup

Examples

```
showClass("kaps")
```

---

kaps.control *Control tuning parameters for "kaps" object*

Description

Tuning parameters for an object "kaps"

Usage

```
kaps.control(pre.pt = list(), scope = list(),
  rho = 0, V = 5, ncl = 1, lower.limit = 0, upper.limit = 100,
  shortcut = TRUE, N.perm = 9999, N.boot = 200, alpha = 0.05,
  splits = c("logrank", "exact"),
  correct = c("Adj.Bf", "Bf", "None"),
  p.adjust.methods = c("none", "holm", "hochberg",
    "hommel", "bonferroni",  "BH", "BY", "fdr"))
```

Arguments

`pre.pt`	list parameter that treats pre-specified split candidates. Use the option as list(var = split points), i.e., x = 1:100
`scope`	list parameter that treats pre-determined split range. Use the option as list(var = ranges), i.e., x = c(1,100)
`rho`	scalar parameter that controls the type of logrank test. See survdiff.
`V`	numeric parameter that determines the number of folds in the cross-validation subgroup selection.
`ncl`	integer parameter that determines the number of cores to improve computing power
`lower.limit`	numeric parameter that treats pre-determined overall lower bound. Default is 0.
`upper.limit`	numeric parameter that treats pre-determined overall upper bound. Default is 100.

`shortcut` logical parameter. If shortcut = TRUE, we skip the off-diagonal matrix in pairwise-comparison to reduce computational cost. The default value is TRUE.

`N.perm` numeric parameter that gives the number of permutation samples used in the kaps algorithm. The default value is 9999.

`N.boot` numeric parameter that gives the number of Bootstrap samples used in the bootstrap and permuting kaps algorithm. The default value is 200.

`alpha` numeric parameter that provides a significant level in the process of Bootstrap and permuting algorithm.

`splits` character parameter that detemines the kind of pairwise test. Default is logrank test. At this stage, the option `exact` is not working.

`correct` character parameter to select the criteria for the multiple comparison in the simple permuting kaps algorithm.

`p.adjust.methods`
character parameter to select the criteria for the multiple comparison.

See Also

`kaps`

`kapsNews` *Show the NEWS file of the kaps package*

Description

Show the NEWS file of the kaps package which contains information about updating and bug fixes.

Usage

```
kapsNews()
```

Examples

```
kapsNews()
```

km.curve *Plot Kaplan-Meire survival curves*

Description

Plot a Kaplan-Meire survival curve for terminal nodes or selected subgroups.

Usage

```
km.curve(object,
x.lab = c(0,24,48,72,96,120, 144, 168, 192, 216, 240), lwd = 1.5, ...)
```

Arguments

`object`	an object from `kaps`
`x.lab`	X labels specified as arguments
`lwd`	line width
`...`	other arguments for plot object. See plot for details.

Details

This function provides Kaplan-Meire survival curves with the estimated subgroups by `kaps`.

See Also

`kaps`

plot *Visualize an object "kaps"*

Description

`plot` method for "kaps" with extended facilities. It provides four panels consisting of a scatter plot, a Kaplan-Meier survival curve, an overall p-values, and a plot with the worst-pair p-values against K.

Usage

```
## S4 method for signature 'kaps'
plot(x, y = NA, K, ...)
```

Arguments

x an object from kaps

y the "y" argument is not used in the plot-method for "OutlierDM" object.

K a scalar object that plots the Kaplan-Meire survival curves for the K. If missing, it works with selected K in the model fitting.

... other arguments to the plot.default function can be passed here.

Details

This function generates four plots. The top left panel is the scatterplot of survival times against the selected prognostic factor with the line fitted by local censored regression using locfit. The top right panel is a Kaplan-Meier survival curve for the subgroups selected with the optimal *K*. At the bottom are displayed the plots of the overall and worst-pair p-values against K. The dotted lines indicate thresholds for significance 0.05.The outputs for a specific K can also be printed out with the argument K.

For the sake of the Kaplan-Meire curve with estimated subgroups, in addition, the function km.curve is provided.

See Also

kaps
km.curve

---

predict *Predict new values using the fitted object "kaps".*

Description

This function provides the predicted subgroup or test statistics.

Usage

```
## S4 method for signature 'kaps'
predict(object, newdata, type = c("predict","kaps"))
```

Arguments

object object from kaps.

newdata An optinal argument in which the name of predicted object is located. If omitted, the dataset used in the model fitting is utilized.

type a type of preddiction. If "predict", predict subgroups based on the fitted model. If a type is "kaps", it returns the overall and worst-pair test staiststics for estimated subgroups.

See Also

kaps

print *Print an object "kaps" with specific information about K*

Description

It functions like show but the only difference is the output with the specific information about K.

Usage

```
## S4 method for signature 'kaps'
print(x, K)
```

Arguments

`x` an object from kaps

`K` a scalar object to determine the number of subgroups K. If missing, the estimated subgroup K is selected.

See Also

kaps

show *Show an object "kaps"*

Description

It returns the outputs of the object "kaps" consisting of three parts. The first part displays the model formula with a dataset and the selected number for K. Next, the information regarding the selection of an optimal set of cut-off points is provided. Lastly, the p-values of pairwise two-sample test comparisons among all the pairs of subgroups are provided.

Usage

```
## S4 method for signature 'kaps'
show(object)
```

Arguments

`object` object from kaps.

See Also

kaps

summary — *Summarize an object "kaps"*

Description

This function provides the tabloid information with survival median, 1-, 3-, and 5 years actual suvival time for each partition.

Usage

```
## S4 method for signature 'kaps'
summary(object, K)
```

Arguments

`object` object with the class `kaps`.

`K` scalar object to determine the number of subgroups K. If missing, the estimated subgroup K is selected.

See Also

`kaps`

toy — *toy example*

Description

toy dataset

Usage

```
data(toy)
```

Value

`meta` covariate variable that describes the number of metastatic lymph nodes

`status` censoring status

`time` time to event

Index

∗Topic **classes**
kaps-class, 6
∗Topic **datasets**
toy, 12
∗Topic **kaps**
count.mindat, 2
kaps, 3
kaps.control, 7
∗Topic **methods**
kapsNews, 8
km.curve, 9
plot, 9
predict, 10
print, 11
show, 11
summary, 12
∗Topic **package**
kaps-package, 2

count.mindat, 2, *5*

Formula, *3, 4*

kaps, *2, 3*, 3, *8–12*
kaps-class, 6
kaps-package, 2
kaps.control, *4, 5*, 7
kapsNews, 8
km.curve, 9, *10*

plot, *5*, *9*, 9
plot,kaps,missing-method *(*plot*)*, 9
plot,kaps-method *(*plot*)*, 9
plot.default, *10*
predict, *5*, 10
predict,kaps-method *(*predict*)*, 10
print, *5*, 11
print,kaps-method *(*print*)*, 11

show, *5*, 11
show,kaps-method *(*show*)*, 11
summary, *5*, 12
summary,kaps-method *(*summary*)*, 12
Surv, *3, 4*
survdiff, *7*

toy, 12