Package 'dynamichazard'

December 28, 2016

Type Package

Title Dynamic Hazard Models using State Space Models

Version 0.1.0

Description Contains functions that lets you fit dynamic hazard models with binary outcomes using state space models. The methods are originally described in Fahrmeir (1992) <doi:10.1080/01621459.1992.10475232> and Fahrmeir (1994) <doi:10.1093/biomet/81.2.317>. The functions also provide an extension hereof where the Extended Kalman filter is replaced by an Unscented Kalman filter. Models are fitted with the regular coxph() like formula.

License GPL-2

LazyData TRUE

Imports Rcpp (>= 0.12.6), stats, graphics, utils, parallel, stringr, survival

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 5.0.1

Suggests testthat, knitr, rmarkdown, timereg, captioner, biglm, httr, mgcv, shiny, formatR

VignetteBuilder knitr

BugReports https://github.com/boennecd/dynamichazard/issues

SystemRequirements C++11

URL https://github.com/boennecd/dynamichazard

NeedsCompilation yes

Author Benjamin Christoffersen [cre, aut],
Alan Miller [ctb],
Anthony Williams [ctb],
Boost developers [ctb, cph],
R-core [ctb, cph]

Maintainer Benjamin Christoffersen <boennecd@gmail.com>

Repository CRAN

Date/Publication 2016-12-28 17:49:25

R topics documented:

`ddFixed` *Function used in formula of* `ddhazard` *for time-invariant effects*

Description

Function used in formula of `ddhazard` for time-invariant effects

Usage

```
ddFixed(object)
```

Arguments

`object`	Expression that would be used in formula. E.g. `x` or `poly(x, degree = 3)`

`ddhazard` *Function to fit dynamic discrete hazard models*

Description

Function to fit dynamic discrete hazard models using state space models

Usage

```
ddhazard(formula, data, model = "logit", by, max_T, id, a_0, Q_0, Q = Q_0,
  order = 1, control = list(), verbose = F)
```

Arguments

formula	coxph like formula with Surv(tstart, tstop, event) on the left hand site of ~
data	Data frame or environment containing the outcome and co-variates
model	"logit", "exp_trunc_time_w_jump", "exp_trunc_time", "exp_bin" or "exp_combined" for the discrete time function using the logistic link function in the first case or for the continuous time model with different estimation method in the four latter cases (see the ddhazard for details on the methods)
by	Interval length of the bins in which parameters are fixed
max_T	End of the last interval. The last stop time with an event is selected if the parameter is omitted
id	Vector of ids for each row of the in the design matrix
a_0	Vector a_0 for the initial coefficient vector for the first iteration (optional). Default is estimates from static model (see static_glm)
Q_0	Covariance matrix for the prior distribution
Q	Initial covariance matrix for the state equation
order	Order of the random walk
control	List of control variables (see details below)
verbose	TRUE if you want status messages during execution

Details

This function can be used to estimate a binary regression where the regression parameters follows a given order random walk. The order is specified by the order argument. 1. and 2. order random walks is implemented. The regression parameters are updated at time by, 2by, ..., max_T. See the vignette 'ddhazard' for more details

The Extended Kalman filter or Unscented Kalman filter needs an initial co-variance matrix Q_0 and state vector a_0. An estimate from a time-invariant model is provided for a_0 if it is not supplied (the same model you would get from static_glm function). A diagonal matrix with large entries is recommended for Q_0. What is large dependents on the data set and model. Further, a variance matrix for the first iteration Q is needed. It is recommended to select diagonal matrix with low values for the latter. The Q, a_0 and optionally Q_0 is estimated with an EM-algorithm

The model is specified through the model argument. Currently, 'logit' and 'exponential' is available. The former uses an logistic model where outcomes are binned into the intervals. Be aware that there can be loss of information due to binning. It is key for the logit model that the id argument is provided if individuals in the data set have time varying co-variates. The latter model uses an exponential model for the arrival times where there is no loss information due to binning

It is recommended to see the Shiny app demo for this function by calling ddhazard_app()

Value

A list with class fahrmeier_94. The list contains:

formula	The passed formula

state_vecs	2D matrix with the estimated state vectors (regression parameters) in each bin
state_vars	3D array with smoothed variance estimates for each state vector
lag_one_cov	3D array with lagged correlation matrix for each for each change in the state vector. Only present wh
n_risk	The number of observations in each interval
times	The interval borders
risk_set	The object from get_risk_obj if saved
data	The data argument if saved
order	Order of the random walk
F_	Matrix with that map transition from one state vector to the next
method	Method used in the E-step
est_Q_0	TRUE if Q_0 was estimated in the EM-algorithm
hazard_func	Hazard function
hazard_first_deriv	First derivative of the hazard function with respect to the linear predictor

Control

The control argument allows you to pass a list to select additional parameters. See the vignette 'ddhazard' for more information on hyper parameters. Unspecified elements of the list will yield default values

method	Set to the method to use in the E-step. Either "EKF" for the Extended Kalman Filter or "UKF
LR	Learning rate for the Extended Kalman filter
NR_eps	Tolerance for the Extended Kalman filter. Default is NULL which means that no extra iteratio
alpha	Hyper parameter α in the Unscented Kalman Filter
beta	Hyper parameter β in the Unscented Kalman Filter
kappa	Hyper parameter κ in the Unscented Kalman Filter
n_max	Maximum number of iteration in the EM-algorithm
eps	Tolerance parameter for the EM-algorithm
est_Q_0	TRUE if you want the EM-algorithm to estimate Q_0. Default is FALSE
save_risk_set	TRUE if you want to save the list from get_risk_obj used to estimate the model. It may be
save_data	TRUE if you want to save the list data argument. It may be needed for later call to residual
ridge_eps	Penalty term added to the diagonal of the covariance matrix of the observational equation in
fixed_terms_method	The method used to estimate the fixed effects. Either 'M_step' or 'E_step' for estimation
Q_0_term_for_fixed_E_step	The diagonal value of the initial covariance matrix, Q_0, for the fixed effects if fixed effects
eps_fixed_parems	Tolerance used in the M-step of the Fisher's Scoring Algorithm for the fixed effects

References

Fahrmeir, Ludwig. *Dynamic modelling and penalized likelihood estimation for discrete time survival data*. Biometrika 81.2 (1994): 317-330.

Durbin, James, and Siem Jan Koopman. *Time series analysis by state space methods*. No. 38. Oxford University Press, 2012.

See Also

plot, residuals, predict, static_glm, ddhazard_app

ddhazard_app *A Shiny app to illustrates model and method*

Description

Runs a shiny app where you try different model specifications on simulated data

Usage

```
ddhazard_app()
```

Examples

```
## Not run:
dynamichazard::ddhazard_app()

## End(Not run)
```

get_risk_obj *Get the risk set at each bin over an equal distance grid*

Description

Get the risk set at each bin over an equal distance grid

Usage

```
get_risk_obj(Y, by, max_T, id, is_for_discrete_model = T)
```

Arguments

Y	Vector of outcome variable
by	Length of each bin
max_T	Last observed time
id	Vector with ids where entries match with outcomes Y
is_for_discrete_model	TRUE/FALSE for whether the model outcome is discrete. For example, a logit model is discrete whereas what is coined an exponential model in this package is a dynamic model

Value

A list with the following elements:

risk_sets	List of lists with one for each bin. Each of the sub lists have indices that corresponds to the entrie
min_start	Start time of the first bin
I_len	Length of each bin
d	Number of bins
is_event_in	Indices for which bin an observation Y is an event. -1 if the individual does not die in any of the b
is_for_discrete_model	Value of is_for_discrete_model argument

---

get_survival_case_weigths_and_data

Static GLM fit for survival models

Description

Function used to get design matrix and weights for a static fit for survivals models where observations are binned into intervals

Usage

```
get_survival_case_weigths_and_data(formula, data, by, max_T, id, init_weights,
  risk_obj, use_weights = T, is_for_discrete_model = T, c_outcome = "Y",
  c_weights = "weights", c_end_t = "t")
```

Arguments

formula	coxph like formula with Surv(tstart, tstop, event) on the left hand site of ~
data	Data frame or environment containing the outcome and co-variates
by	Length of each intervals that cases are binned into
max_T	The end time of the last bin
id	The id for each row in data. This is important when variables are time varying
init_weights	Weights for the rows data. Useful with skewed sampling and will be used when computing the final weights
risk_obj	A pre-computed result from a get_risk_obj. Will be used to skip some computations
use_weights	TRUE if weights should be used. See details
is_for_discrete_model	TRUE if the model is for a discrete hazard model like the logistic model. Affects how deaths are included when individuals have time varying coefficients
c_outcome, c_weights, c_end_t	Alternative names to use for the added columns described in the return section. Useful if you already have a column named Y, t or weights

Details

This function is used to get the data frame for e.g. a `glm` fit that is comparable to a `ddhazard` fit in the sense that it is a static version. For example, say that we bin our time periods into `(0,1]`, `(1,2]` and `(2,3]`. Next, consider an individual who dies at time 2.5. He should be a control in the the first two bins and should be a case in the last bin. Thus the rows in the final data frame for this individual is `c(Y = 1, ..., weights = 1)` and `c(Y = 0, ..., weights = 2)` where `Y` is the outcome, `...` is the co-variates and `weights` is the weights for the regression. Consider another individual who does not die and we observe him for all three periods. Thus, he will yield one row with `c(Y = 0, ..., weights = 3)`

This function use similar logic as the `ddhazard` for individuals with time varying co-variates (see the vignette "ddhazard" for details)

If `use_weights = FALSE` then the two individuals will yield three rows each. The first individual will have `c(Y = 0, t = 1, ..., weights = 1)`, `c(Y = 0, t = 2, ..., weights = 1)`, `c(Y = 1, t = 3, ..., weights = 1)` while the latter will have three rows `c(Y = 0, t = 1, ..., weights = 1)`, `c(Y = 0, t = 2, ..., weights = 1)`, `c(Y = 0, t = 3, ..., weights = 1)`. This kind of data frame is useful if you want to make a fit with e.g. `gam` function in the `mgcv` package as described en Tutz et. al (2016) (see reference)

Value

Returns a data frame with the design matrix from the formula where the following is added (column names will differ if you specified them): column `Y` for the binary outcome, column `weights` for weights of each row and additional rows if applicable. A column `t` is added for the stop time of the bin if `use_weights = FALSE`

References

Tutz, Gerhard, and Matthias Schmid. *Nonparametric Modeling and Smooth Effects*. Modeling Discrete Time-to-Event Data. Springer International Publishing, 2016. 105-127.

See Also

`ddhazard`, `static_glm`

`plot.fahrmeier_94` *Plots for* `ddhazard`

Description

Plot to illustrate the estimate state space variables from a `ddhazard` fit

Usage

```
## S3 method for class 'fahrmeier_94'
plot(x, xlab = "Time", ylab = "Hazard",
  type = "cov", plot_type = "l", cov_index, ylim, col = "black",
  add = F, do_alter_mfcol = T, ...)
```

Arguments

x	Result of ddhazard call
xlab, ylab, ylim, col	Arguments to override defaults set in the function
type	Type of plot. Currently, only "cov" is available for plot of the state space parameters
plot_type	The type argument passed to plot
cov_index	The index (indices) of the state space parameter(s) to plot
add	FALSE if you want to make a new plot
do_alter_mfcol	TRUE if the function should alter par(mfcol) in case that cov_index has more than one element
...	Arguments passed to plot or lines depending on the value of add

Details

Creates a plot of state variables or adds state variables to a plot with indices cov_index. Pointwise 1.96 std. confidence intervals are provided with the smoothed co-variance matrices from the fit

---

plot.fahrmeier_94_SpaceErrors

State space error plot

Description

Plot function for state space errors from ddhazard fit

Usage

```
## S3 method for class 'fahrmeier_94_SpaceErrors'
plot(x, mod, cov_index = NA,
  t_index = NA, p_cex = par()$cex * 0.2, pch = 16,
  ylab = "Std. state space error", x_tick_loc = NA, x_tick_mark = NA,
  xlab = "Time", ...)
```

Arguments

x	Result of residuals for state space errors
mod	The ddhazard result used in the residuals call
cov_index	The indices of state vector errors to plot. Default is to use all which is likely what you want if the state space errors are standardized
t_index	The bin indices to plot. Default is to use all bins
p_cex	cex argument for the points
pch, ylab, xlab	Arguments to override defaults set in the function

x_tick_loc, x_tick_mark	at and labels arguments passed to axis
...	Arguments passed to plot

---

predict.fahrmeier_94 *Predict function for the result of* ddhazard

---

Description

Predict function for the result of ddhazard

Usage

```
## S3 method for class 'fahrmeier_94'
predict(object, new_data, type = c("response", "term"),
  tstart = "start", tstop = "stop", use_parallel = F, sds = F,
  max_threads = getOption("ddhazard_max_threads"), ...)
```

Arguments

object	Result of a ddhazard call
new_data	New data to base predictions on
type	Either "response" for predicted probability of death or "term" for predicted terms in the linear predictor
tstart	Name of the start time column in new_data. It must corresponds to tstart used in the Surv(tstart, tstop, event) in the formula passed to ddhazard
tstop	same as tstart for the stop argument
use_parallel	TRUE if computation for type = "response" should be computed in parallel with the parallel package
sds	TRUE if point wise standard deviation should be computed. Convenient if you use functions like ns and you only want one term per term in the right hand site of the formula used in ddhazard
max_threads	Maximum number of threads to use. -1 if it should be determined by a call to detectCores
...	Not used

Term

The result of type = "term" is a list with the following elements

terms	Is a 3D array. The first dimension is the number of bins, the second dimension is rows in new_data and the la
sds	Similar to terms for the point wise confidence intervals using the smoothed co-variance matrices
fixed_terms	Vector of the fixed effect terms for each observation

Response

The result of `type = "response"` is a list with the elements below. The function check if there are columns in `new_data` which's names match `tstart` and `tstop`. If not, then each row in new data will get a predicted probability of dying in every bin.

`fits`	Fitted probability of dying
`istart`	Vector with the index of the first bin the elements in `fits` is in
`istop`	Vector with the index of the last bin the elements in `fits` is in

---

`residuals.fahrmeier_94`

Residuals for `ddhazard`

Description

Residuals function for the result of a `ddhazard` fit

Usage

```
## S3 method for class 'fahrmeier_94'
residuals(object, type = c("std_space_error",
  "space_error", "pearson", "raw"), data = NULL, ...)
```

Arguments

`object`	Result of `ddhazard` call
`type`	Type of residuals. Four possible values: `"std_space_error"`, `"space_error"`, `"pearson"` and `"raw"`. See the sections below for details
`data`	Data frame with data for Pearson or raw residuals
`...`	Not used

Pearson and raw residuals

Is the result of a call with a `type` argument of either `"pearson"` or `"raw"` for Pearson residuals or raw residuals. Returns a list with class `"fahrmeier_94_res"` with the following elements

`residuals`	List of residuals for each bin. Each element of the list contains a 2D array where the rows corresponds to the pas
`type`	The type of residual

State space errors

Is the result of a call with a `type` argument of either `"std_space_error"` or `"space_error"`. The former is for standardized residuals while the latter is non-standardized. Returns a list with class `"fahrmeier_94_SpaceErrors"` with the following elements

residuals	2D array with either standardized or non-standardized state space errors. The row are bins and the columns ar
standardize	TRUE if standardized state space errors
Covariances	3D array with the smoothed co-variance matrix for each set of the state space errors

---

static_glm *Function to make a static glm fit*

Description

Function to make a static glm fit

Usage

```
static_glm(formula, data, by, max_T, id, family = "logit", model = F,
  weights, risk_obj = NULL, ...)
```

Arguments

formula	coxph like formula with Surv(tstart, tstop, event) on the left hand site of ~
data	Data frame or environment containing the outcome and co-variates
by	Length of each intervals that cases are binned into
max_T	The end time of the last bin
id	The id for each row in data. This is important when variables are time varying
family	"logit" or "exponential" for the static equivalent model of ddhazard
model	TRUE if you want to save the design matrix used in glm
weights	weights if a skewed sample or similar is used
risk_obj	A pre-computed result from a get_risk_obj. Will be used to skip some computations
...	arguments passed to glm

Details

Method to fit a static model corresponding to a ddhazard fit. The method uses weights to ease the memory requirements. See get_survival_case_weigths_and_data for details on weights

Value

The returned list from the glm call

Index