Package 'blapsr'

August 20, 2022

Type Package

Title Bayesian Inference with Laplace Approximations and P-Splines

Version 0.6.1

Depends R (>= 3.6.0), survival (>= 2.44.1)

Maintainer Oswaldo Gressani <oswaldo_gressani@hotmail.fr>

Description Laplace approximations and penalized B-splines are combined
for fast Bayesian inference in latent Gaussian models. The routines can be
used to fit survival models, especially proportional hazards and promotion
time cure models (Gressani, O. and Lambert, P. (2018)
<doi:10.1016/j.csda.2018.02.007>). The Laplace-P-spline methodology can also
be implemented for inference in (generalized) additive models
(Gressani, O. and Lambert, P. (2021) <doi:10.1016/j.csda.2020.107088>).
See the associated website for more information and examples.

URL <https://www.blapsr-project.org/>

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.2.1

Imports coda (>= 0.19.3), graphics(>= 3.6.0), MASS (>= 7.3.51), Matrix
(>= 1.2.17), RSpectra (>= 0.16.0), sn (>= 1.5.4), stats, utils
(>= 3.6.0)

Suggests knitr (>= 1.26), rmarkdown (>= 1.14), testthat (>= 2.3.1)

VignetteBuilder knitr

NeedsCompilation no

Author Oswaldo Gressani [aut, cre] (Author),
Philippe Lambert [aut, ths] (Co-author and thesis advisor)

Repository CRAN

Date/Publication 2022-08-20 13:10:02 UTC

R **topics documented:**

---

amlps   *Bayesian additive partial linear modeling with Laplace-P-splines.*

Description

Fits an additive partial linear model to data using an approximate Bayesian inference technique based on penalized regression splines and Laplace approximations. Smooth additive terms are specified as a linear combination of of a large number of cubic B-splines. To counterbalance the roughness of the fit, a discrete penalty on neighboring spline coefficients is imposed in the spirit of Eilers and Marx (1996). The error of the model is assumed to be Gaussian with zero mean and finite variance.

The optimal amount of smoothing is determined by a grid-based exploration of the posterior penalty space when the number of smooth terms is small to moderate. When the dimension of the penalty

space is large, the optimal smoothing parameter is chosen to be the value that maximizes the (log-)posterior of the penalty vector.

Usage

```
amlps(formula, data, K = 30, penorder = 2, cred.int = 0.95)
```

Arguments

formula
: A formula object where the ~ operator separates the response from the covariates of the linear part z1,z2,.. and the smooth terms. A smooth term is specified by using the notation sm(.). For instance, the formula y ~ z1+sm(x1)+sm(x2) specifies an additive model of the form *E(y)=b0+b1z1+f1(x1)+f2(x2)*, where *b0, b1* are the regression coefficients of the linear part and *f1(.)* and *f2(.)* are smooth functions of the continuous covariates *x1* and *x2* respectively.

data
: Optional. A data frame to match the variable names provided in formula.

K
: A positive integer specifying the number of cubic B-spline functions in the basis used to model the smooth terms. Default is K = 30 and allowed values are 15 <= K <= 60. The same basis dimension is used for each smooth term in the model. Also, the computational cost to fit the model increases with K.

penorder
: The penalty order used on finite differences of the coefficients of contiguous B-splines. Can be either 2 for a second-order penalty (the default) or 3 for a third-order penalty.

cred.int
: The level of the pointwise credible interval to be computed for the coefficients in the linear part of the model.

Details

The B-spline basis used to approximate a smooth additive component is computed with the function cubicbs. The lower (upper) bound of the B-spline basis is taken to be the minimum (maximum) value of the covariate associated to the smooth. For identifiability purposes, the B-spline matrices (computed over the observed covariates) are centered. The centering consists is subtracting from each column of the B-spline matrix, the corresponding column average of another B-spline matrix computed on a fine grid of equidistant values in the domain of the smooth term.

A hierarchical Gamma prior is imposed on the roughness penalty vector and Jeffreys' prior is imposed on the precision of the error. A Newton-Raphson algorithm is used to compute the posterior mode of the (log-)posterior penalty vector. The latter algorithm uses analytically derived versions of the gradient and Hessian. When the number of smooth terms in the model is smaller or equal to 4, a grid-based strategy is used for posterior exploration of the penalty space. Above that threshold, the optimal amount of smoothness is determined by the posterior maximum value of the penalty vector. This strategy allows to keep the computational burden to fit the model relatively low and to conserve good statistical performance.

Value

An object of class amlps containing several components from the fit. Details can be found in amlps.object. Details on the output printed by amlps can be found in print.amlps. Fitted smooth terms can be visualized with the plot.amlps routine.

Author(s)

Oswaldo Gressani <oswaldo_gressani@hotmail.fr>.

References

Eilers, P.H.C. and Marx, B.D. (1996). Flexible smoothing with B-splines and penalties. *Statistical Science*, **11**(2): 89-121.

Fan, Y. and Li, Q. (2003). A kernel-based method for estimating additive partially linear models. *Statistica Sinica*, **13**(3): 739-762.

Gressani, O. and Lambert, P. (2018). Fast Bayesian inference using Laplace approximations in a flexible promotion time cure model based on P-splines. *Computational Statistical & Data Analysis* **124**: 151-167.

Opsomer, J. D. and Ruppert, D. (1999). A root-n consistent backfitting estimator for semiparametric additive modeling. *Journal of Computational and Graphical Statistics*, **8**(4): 715-732.

See Also

cubicbs, amlps.object, print.amlps, plot.amlps

Examples

```
### Classic simulated data example (with simgamdata)

set.seed(17)
sim.data <- simgamdata(setting = 2, n = 200, dist = "gaussian", scale = 0.4)
data <- sim.data$data  # Simulated data frame

# Fit model
fit <- amlps(y ~ z1 + z2 + sm(x1) + sm(x2), data = data, K = 15)
fit
```

---

amlps.object   *Object resulting from the fit of an additive partial linear model.*

Description

An object returned by the amlps function consists in a list with various components related to the fit of an additive partial linear model with the Laplace-P-spline approach.

Value

An amlps object has the following elements:

formula	The formula of the additive model.
n	Sample size.

q	Total number of smooth terms.
K	Number of B-spline basis functions used for the fit.
penalty.order	Chosen penalty order.
latfield.dim	The dimension of the latent field. This is equal to the sum of the number of B-spline coefficients and the number of regression parameters related to the co-variates in the linear part.
linear.coeff	Estimated linear regression coefficients. This is a matrix containing the posterior point estimate, standard deviation and lower/upper bounds of the credible interval.
spline.estim	The estimated B-spline coefficients. This is a list with q vectors of size K-1 representing the estimated B-spline amplitudes for each smooth term.
edf	Estimated effective degrees of freedom for each latent field variable.
Approx.signif	A matrix returning the observed test statistic and p-value for the approximate significance of smooth terms.
EDf	The estimated effective degrees of freedom of the smooth terms.
EDfHPD.95	95% HPD interval for the degrees of freedom of the smooth terms.
ED	The estimated degrees of freedom of the additive model.
sd.error	The estimated standard deviation of the error.
vmap	The maximum a posteriori of the (log) posterior penalty vector.
Cov.vmap	Covariance matrix of the (log) posterior penalty vector evaluated at vmap.
pen.family	The family of the posterior distribution for v. It is either "skew-normal" or "gaussian".
pendist.params	The parameterization for the posterior distribution of v. If the posterior of v belongs to the skew-normal family, then pendist.params is a matrix with as many rows as the number of smooth terms q. Each row contains the location, scale and shape parameter of the skew-normal distribution. If the posterior of v belongs to the Gaussian family, then pendist.params is a vector of length q, corresponding to vmap.
Covmaximum	The covariance matrix of the latent field evaluated at vmap.
latmaximum	The latent field vector evaluated at vmap.
fitted.values	The fitted response values.
residuals	The response residuals.
r2.adj	The adjusted r-squared of the model indicating the proportion of the data variance explained by the model fit.
data	The data frame.

Author(s)

Oswaldo Gressani <oswaldo_gressani@hotmail.fr>.

See Also

[amlps](), [print.amlps](), [plot.amlps]()

---

coxlps	*Fit a Cox proportional hazards regression model with Laplace-P-splines.*

Description

Fits a Cox proportional hazards regression model for right censored data by combining Bayesian P-splines and Laplace approximations.

Usage

```
coxlps(formula, data, K = 30, penorder = 2, tmax = NULL)
```

Arguments

formula	A formula object where the ~ operator separates the response from the covariates. In a Cox model, it takes the form *response ~ covariates*, where *response* is a survival object returned by the [Surv](#) function of the *survival* package.
data	Optional. A data frame to match the variable names provided in formula.
K	A positive integer specifying the number of cubic B-spline functions in the basis. Default is *K = 30* and allowed values are 10 <= K <= 60.
penorder	The penalty order used on finite differences of the coefficients of contiguous B-splines. Can be either 2 for a second-order penalty (the default) or 3 for a third-order penalty.
tmax	A user-specified value for the upper bound of the B-spline basis. The default is NULL, so that the B-spline basis is specified in the interval *[0, tup]*, where *tup* is the upper bound of the follow-up times. It is required that *tmax > tup*.

Details

The log-baseline hazard is modeled as a linear combination of K cubic B-splines as obtained from [cubicbs](#). The B-spline basis is specified in the interval *[0, tup]*, where *tup* is the upper bound of the follow-up times, i.e. the largest observed follow-up time. Following Jullion and Lambert (2007), a robust Gamma prior is imposed on the roughness penalty parameter. A grid-based approach is used to explore the posterior penalty space and the resulting quadrature points serve to compute the approximate (joint) marginal posterior of the latent field vector. Point and set estimates of latent field elements are obtained from a finite mixture of Gaussian densities. The routine centers the columns of the covariate matrix around their mean value for numerical stability.

Value

An object of class coxlps containing various components from the fit. Details can be found in coxlps.object. Plot of estimated smooth hazard and survival curves can be obtained using [plot.coxlps](#). If required, estimated baseline quantities on specific time values can be obtained with coxlps.baseline.

Author(s)

Oswaldo Gressani <oswaldo_gressani@hotmail.fr>.

References

Cox, D.R. (1972). Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)* **34**(2): 187-202.

Gressani, O. and Lambert, P. (2018). Fast Bayesian inference using Laplace approximations in a flexible promotion time cure model based on P-splines. *Computational Statistical & Data Analysis* **124**: 151-167.

Jullion, A. and Lambert, P. (2007). Robust specification of the roughness penalty prior distribution in spatially adaptive Bayesian P-splines models. *Computational Statistical & Data Analysis* **51**(5): 2542-2558.

See Also

Surv, coxph, simsurvdata

Examples

```
### Example 1 (Simulated survival data)

set.seed(3)

# Simulate survival data  with simsurvdata
betas <- c(0.13, 0.52, 0.30)
simul <- simsurvdata(a = 3.8, b = 2.2, n = 250, betas = betas , censperc = 20)
simul
simdat <- simul$survdata
plot(simul) # Plot survival data

# Estimation with coxlps
fit <- coxlps(Surv(time, delta) ~ x1 + x2 + x3, data = simdat, K = 15)
# Compare coxlps and coxph
fit
summary(coxph(Surv(time, delta) ~ x1 + x2 + x3, data = simdat))

# Fitted baseline survival vs target
plot(fit, h0 = FALSE, cred.int = 0.95, overlay.km = TRUE)
domt <- seq(0, 4, length = 100)
lines(domt, simul$S0(domt), type = "l", col = "red")
legend("topright", col=c("black", "blue", "red"), lty = rep(1,3),
      c("Bayesian LPS", "Kaplan-Meier", "Target"), cex = 0.8, bty = "n")

### Example 2 (Kidney transplant data)

data(kidneytran)
Surv.obj <- Surv(kidneytran$time, kidneytran$delta)
fit <- coxlps(Surv.obj ~ age + gender + race, data = kidneytran)
```

```
coxphfit <- coxph(Surv.obj ~ age + gender + race, data = kidneytran)
## Compare coxph and coxlps results
summary(coxphfit)
fit
## Plot Kaplan-Meier curve vs Laplace-P-spline fit
plot(fit, h0 = FALSE, overlay.km = TRUE, plot.cred = FALSE)


### Example 3 (Laryngeal cancer data)

data(laryngeal)
fit <- coxlps(Surv(time, delta) ~ age + diagyr + as.factor(stage),
              data = laryngeal)
coxphfit <- coxph(Surv(time, delta) ~ age + diagyr + as.factor(stage),
                  data = laryngeal)
## Compare coxph and coxlps results
summary(coxphfit)
fit
## Plot Kaplan-Meier curve vs Laplace-P-spline fit
plot(fit, h0 = FALSE, overlay.km = TRUE, plot.cred = FALSE)
```

---

coxlps.baseline   *Extract estimated baseline quantities from a fit with coxlps.*

Description

The routine takes as input an object of class 'coxlps' and computes point estimates and credible
intervals for the baseline hazard and survival on a user-specified time vector.

Usage

```
coxlps.baseline(object, time = NULL, compute.cred = TRUE, cred.int = 0.95,
                verbose = TRUE)
```

Arguments

object	An object of class 'coxlps'.
time	A vector of time values on which to compute the estimated baseline quantities. Each component of 'time' must be between 0 and the largest observed follow-up time. If time is 'NULL' (the default), then only the baseline median lifetime (if available) is computed.
compute.cred	Should the credible intervals be computed? Default is TRUE.
cred.int	The level for an approximate pointwise credible interval to be computed for the baseline hazard and survival curves. Default is 0.95.
verbose	Should the table of estimated values be printed to console? Default is TRUE.

Value

A list with the following components:

fit.time
: A matrix with point and set estimates of the baseline hazard and survival curves for values provided in 'time'. Only available if 'time' is not 'NULL'. Column *Time* summarizes the provided values in 'time'. Columns named *h0*, *S0*, are the point estimates of the baseline hazard and baseline survival respectively. *low* and *up* give the lower and upper bound respectively of the approximate pointwise credible interval.

median.lifetime
: The estimated baseline median lifetime.

cred.int
: The chosen level to construct credible intervals.

Author(s)

Oswaldo Gressani <oswaldo_gressani@hotmail.fr>.

Examples

```
## Simulate survival data
set.seed(2)
betas <- c(0.15, 0.82, 0.41) # Regression coefficients
data <- simsurvdata(a = 1.8, b = 2, n = 300, betas = betas, censperc = 15)
simdat <- data$survdata

# Fit model
fit <- coxlps(Surv(time, delta) ~ x1 + x2 + x3, data = simdat, K = 20)
coxlps.baseline(fit, time = seq(0, 2, by = 0.5), cred.int = 0.90)
```

---

coxlps.object   *Object from a Cox proportional hazards fit with Laplace-P-splines.*

Description

An object returned by the [coxlps](#) function consists in a list with various components related to the fit of a Cox model using the Laplace-P-spline methodology.

Value

A coxlps object has the following elements:

formula
: The formula of the Cox model.

K
: Number of B-spline basis functions used for the fit.

penalty.order
: Chosen penalty order.

latfield.dim	The dimension of the latent field. This is equal to the sum of the number of B-spline coefficients and the number of regression parameters related to the co-variates.
n	Sample size.
num.events	The number of events that occurred.
event.times	The standardized event times, i.e. if t denotes the original time scale, then event.times = t / sd(t), where sd is the standard deviation.
tup	The upper bound of the follow-up, i.e. max(event.times).
sd.time	The standard deviation of the event times in original scale.
event.indicators	
	The event indicators.
regcoeff	Posterior estimates of the regression coefficients. *coef* gives the point estimate, *sd.post* gives the posterior standard deviation, *z* is the Wald test statistic, *lower .95* and *upper .95* the posterior approximate 95% quantile-based credible interval.
penalty.vector	The selected grid of penalty values.
vmap	The maximum a posteriori of the (log) penalty parameter.
spline.estim	The estimated B-spline coefficients.
edf	Estimated effective degrees of freedom for each latent field variable.
ED	The effective model dimension.
Covthetamix	The posterior covariance matrix of the B-spline coefficients.
X	The matrix of covariate values.
loglik	The log-likelihood evaluated at the posterior latent field estimate.
p	Number of parametric coefficients in the model.
AIC.p	The AIC computed with the formula *-2*loglik+2*p*, where *p* is the number of parametric coefficients.
AIC.ED	The AIC computed with the formula *-2*loglik+2*ED*, where *ED* is the effective model dimension.
BIC.p	The BIC computed with the formula *-2*loglik+p*log(ne)*, where *p* is the number of parametric coefficients and *ne* the number of events.
BIC.ED	The BIC computed with the formula *-2*loglik+ED*log(ne)*, where *ED* is the effective model dimension and *ne* the number of events.

Author(s)

Oswaldo Gressani <oswaldo_gressani@hotmail.fr>.

See Also

[coxlps](), [coxlps.baseline]()

---

cubicbs | *Construct a cubic B-spline basis.*

Description

Computation of a cubic B-spline basis matrix.

Usage

```
cubicbs(x, lower, upper, K)
```

Arguments

x	A numeric vector containing the values on which to evaluate the B-spline basis.
lower, upper	The lower and upper bounds of the B-spline basis domain. Must be finite with lower < upper.
K	A positive integer specifying the number of B-spline functions in the basis.

Value

An object of class cubicbs for which print and plot methods are available. The cubicbs class consists of a list with the following components:

x	A numeric vector on which the basis is evaluated.
lower, upper	The lower and upper bounds of the basis domain.
K	The number of cubic B-spline functions in the basis.
knots	The knot sequence to build the basis.
nknots	Total number of knots.
dimbasis	The dimension of the B-spline basis matrix.
Bmatrix	The B-spline basis matrix.

The print method summarizes the B-spline basis and the plot method gives a graphical representation of the basis with dashed vertical lines indicating knot placement and blue ticks the coordinates of x.

Author(s)

Oswaldo Gressani <oswaldo_gressani@hotmail.fr>.

The core algorithm of the cubicbs function owes much to a code written by Phlilippe Lambert.

References

Eilers, P.H.C. and Marx, B.D. (1996). Flexible smoothing with B-splines and penalties. *Statistical Science*, **11**(2): 89-121.

Examples

```
lb <- 0  # Lower bound
ub <- 1  # Upper bound
xdom <- runif(100, lb, ub) # Draw uniform values between lb and ub
Bsmat <- cubicbs(xdom, lb, ub, 25) # 100 x 25 B-spline matrix
Bsmat
plot(Bsmat) # Plot the basis
```

---

curelps   *Promotion time cure model with Laplace P-splines.*

Description

Fits a promotion time cure model with the Laplace-P-spline methodology. The routine can be applied to survival data for which a plateau is observed in the Kaplan-Meier curve. In this case, the follow-up period is considered to be sufficiently long to intrinsically account for long-term survivors and hence a cured fraction. The user can separately specify the model covariates influencing the cure probability (long-term survival) and the population hazard dynamics (short-term survival).

Usage

```
curelps(formula, data, K = 30, penorder = 2, tmax = NULL, constr = 6)
```

Arguments

formula	A formula object where the ~ operator separates the response from the covariates. In a promotion time cure model, it takes the form *response ~ covariates*, where *response* is a survival object returned by the [Surv](#) function of the *survival* package. The model covariates influencing the long-term survival can be specified in the function lt(.) separated by '+', while the covariates affecting the short-term survival can be specified in st(.). For instance, a promotion time cure model with covariates specified as lt(x1+x2)+st(x1), means that x1 will jointly influence the long- and short-term survival, while x2 will only influence the long-term survival.
data	Optional. A data frame to match the variable names provided in formula.
K	A positive integer specifying the number of cubic B-spline functions in the basis. Default is K = 30 and allowed values are 10 <= K <= 60.
penorder	The penalty order used on finite differences of the coefficients of contiguous B-splines. Can be either 2 for a second-order penalty (the default) or 3 for a third-order penalty.
tmax	A user-specified value for the upper bound of the B-spline basis. The default is NULL, so that the B-spline basis is specified in the interval *[0, tup]*, where *tup* is the upper bound of the follow-up times. It is required that *tmax > tup*.
constr	Constraint imposed on last B-spline coefficient (default is 6).

Details

The log-baseline hazard is modeled as a linear combination of K cubic B-splines as obtained from `cubicbs`. A robust Gamma prior is imposed on the roughness penalty parameter. A grid-based approach is used to explore the posterior penalty space and the resulting quadrature points serve to compute the approximate (joint) posterior of the latent field vector. Point and set estimates of latent field elements are obtained from a finite mixture of Gaussian densities. The routine centers the columns of the covariate matrix around their mean value for numerical stability. See `print.curelps` for a detailed explanation on the output printed by the curelps function.

Value

An object of class `curelps` containing various components from the promotion time cure model fit. Details can be found in `curelps.object`. Estimates on the baseline survival, population survival (for a chosen covariate profile) and cure probability can be obtained with the `plot.curelps` and `curelps.extract` routines.

Author(s)

Oswaldo Gressani <oswaldo_gressani@hotmail.fr>.

References

Cox, D.R. (1972). Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)* **34**(2): 187-202.

Bremhorst, V. and Lambert, P. (2016). Flexible estimation in cure survival models using Bayesian P-splines. *Computational Statistical & Data Analysis* **93**: 270-284.

Gressani, O. and Lambert, P. (2018). Fast Bayesian inference using Laplace approximations in a flexible promotion time cure model based on P-splines. *Computational Statistical & Data Analysis* **124**: 151-167.

Lambert, P. and Bremhorst, V. (2019). Estimation and identification issues in the promotion time cure model when the same covariates influence long- and short-term survival. *Biometrical Journal* **61**(2): 275-289.

See Also

`curelps.object`, `curelps.extract`, `plot.curelps`, `print.curelps`, `Surv`.

Examples

```
## Fit a promotion time cure model on malignant melanoma data

data(melanoma)
medthick <- median(melanoma$thickness)

# Kaplan-Meier estimate to check the existence of a plateau
KapMeier <- survfit(Surv(time,status) ~ 1, data = melanoma)
plot(KapMeier, mark.time = TRUE, mark = 4, xlab = "Time (in years)")
```

```
# Fit with curelps
fit <- curelps(Surv(time , status) ~ lt(thickness + ulcer) +
                    st(thickness + ulcer), data = melanoma, K = 40)
fit

# Cure prediction for median thickness and absence of ulceration
curelps.extract(fit, time = c(2, 4 ,6, 8), curvetype = "probacure",
                cred.int = 0.90, covar.profile = c(medthick, 0, medthick, 0))

# Plot of baseline and population survival functions
opar <- par(no.readonly = TRUE)
par(mfrow = c(1, 2))
# Baseline survival
plot(fit, curvetype = "baseline", plot.cred = FALSE, ylim = c(0,1))
# Population survival
plot(fit, curvetype = "population", covar.profile = c(medthick, 0, medthick, 0),
plot.cred = FALSE, ylim = c(0,1))
par(opar)
```

---

curelps.extract	*Extract estimates of survival functions and cure probability for the promotion time cure model.*

Description

The routine takes as input an object of class `curelps` and computes estimates of the baseline survival curve, the population survival curve and the cure probability on a specified time vector. Approximate pointwise credible intervals are available.

Usage

```
curelps.extract(object, time = NULL, curvetype = c("baseline", "population", "probacure"),
            covar.profile, compute.cred = TRUE, cred.int = 0.95, verbose = TRUE)
```

Arguments

object	An object of class `curelps`.
time	A vector of time values on which to compute the estimates. Each component of `time` must be between 0 and the largest observed follow-up time.
curvetype	The curve on which estimates are computed ; `baseline` (the default) is for the baseline survival, `population` is for the population survival function for a profile of covariates given in `covar.profile`, and `probacure` is for the probability to be cured (for a profile of covariates given in `covar.profile`) given that the subject has survived until time t.

covar.profile	A numeric vector of the same length as the number of covariates in the model. This corresponds to the profile of covariates for which to compute the population survival function and cure probability estimates. The order of the covariates in `covar.profile` is the same as the order specified in `formula` of the `curelps` routine. Each component of `covar.profile` should be in the range of the observed values for the corresponding covariate. If `covar.profile` is left unspecified by the user, the default will be to take the median covariate values.
compute.cred	Should credible intervals be computed? Default is TRUE.
cred.int	The level for an approximate pointwise credible interval. Default is 0.95.
verbose	Should estimates be printed to console?

Value

A list with the following components:

fit.time	Estimates on the time values provided in `time`.
cred.int	The chosen level to construct approximate pointwise credible intervals.
covar.profile	The chosen profile of covariates.

Author(s)

Oswaldo Gressani <oswaldo_gressani@hotmail.fr>.

See Also

[curelps](), [curelps.object](), [plot.curelps](), [print.curelps]().

Examples

```
# Example on phase III clinical trial e1684 on melanoma data

data(ecog1684)

# Kaplan-Meier curve
plot(survfit(Surv(time, status) ~ 1, data = ecog1684), mark.time = TRUE)
fit <- curelps(Surv(time, status) ~ lt(age + trt+ sex) +
            st(age + trt + sex), data = ecog1684, K = 20, penorder = 2)
fit
profile1 <- c(0, 1, 1, 0, 1, 1) # Mean age, trt = IFN, sex = Female.
profile2 <- c(0, 0, 1, 0, 0, 1) # Mean age, trt = control, sex = Female.

# Extract cure probabilities
curelps.extract(fit, time = c(0, 1, 2, 3), curvetype = "probacure",
                covar.profile = profile1, cred.int = 0.90)
curelps.extract(fit, time = c(0, 1, 2, 3), curvetype = "probacure",
                covar.profile = profile2, cred.int = 0.90)
```

---

curelps.object   *Object from a promotion time model fit with Laplace-P-splines.*

Description

An object returned by the [curelps](#) function consists in a list with various components related to the fit of a promotion time cure model using the Laplace-P-spline methodology.

Value

A curelps object has the following elements:

formula	The formula of the promotion time cure model.
K	Number of B-spline basis functions used for the fit.
penalty.order	Chosen penalty order.
latfield.dim	The dimension of the latent field. This is equal to the sum of the number of B-spline coefficients and the number of regression parameters related to the covariates.
event.times	The observed event times.
n	Sample size.
num.events	The number of events that occurred.
tup	The upper bound of the follow up, i.e. max(event.times).
event.indicators	
	The event indicators.
coeff.probacure	
	Posterior estimates of the regression coefficients related to the cure probability (or long-term survival).
coeff.cox	Posterior estimates of the regression coefficients related to the population hazard dynamics (or short-term survival).
vmap	The maximum a posteriori of the (log-)posterior penalty parameter.
vquad	The quadrature points of (log-) posterior penalty parameters used to compute the Gaussian mixture posterior of the latent field vector.
spline.estim	The estimated B-spline coefficients.
edf	Estimated effective degrees of freedom for each latent field variable.
ED	The effective model dimension.
Covtheta.map	The posterior covariance matrix of the B-spline coefficients for a penalty fixed at its maximum posterior value.
Covlatc.map	The posterior covariance matrix of the latent field for a penalty fixed at its maximum posterior value.
X	The covariate matrix for the long-term survival part.
Z	The covariate matrix for the short-term survival part.

loglik	The log-likelihood evaluated at the posterior latent field estimate.
p	Number of parametric coefficients in the model.
AIC.p	The AIC computed with the formula *-2*loglik+2*p*, where *p* is the number of parametric coefficients.
AIC.ED	The AIC computed with the formula *-2*loglik+2*ED*, where *ED* is the effective model dimension.
BIC.p	The BIC computed with the formula *-2*loglik+p*log(ne)*, where *p* is the number of parametric coefficients and *ne* the number of events.
BIC.ED	The BIC computed with the formula *-2*loglik+ED*log(ne)*, where *ED* is the effective model dimension and *ne* the number of events.

Author(s)

Oswaldo Gressani <oswaldo_gressani@hotmail.fr>.

See Also

curelps

---

ecog1684	*Phase III Melanoma clinical trial.*

Description

Melanoma data from the phase III Eastern Cooperative Oncology Group (ECOG) two-arm clinical trial studied in Kirkwood et al. (1996).

Usage

```
data(ecog1684)
```

Format

A data frame with 284 rows and 5 columns.

trt Treatment: 0=control, 1=Interferon alpha-2b (IFN).

time Relapse-free survival (in years).

status 1=death or relapse, 0=censored.

age Age centered to the mean.

sex 0=Male, 1=Female.

Source

<https://CRAN.R-project.org/package=smcure>

References

Kirkwood, J. M., Strawderman, M. H., Ernstoff, M. S., Smith, T. J., Borden, E. C. and Blum, R. H. (1996). Interferon alfa-2b adjuvant therapy of high-risk resected cutaneous melanoma: the Eastern Cooperative Oncology Group Trial EST 1684. *Journal of clinical oncology* **14**(1): 7-17.

Corbiere, F. and Joly, P. (2007). A SAS macro for parametric and semiparametric mixture cure models. *Computer methods and programs in Biomedicine* **85**(2): 173-180.

---

gamlps   *Bayesian generalized additive modeling with Laplace-P-splines.*

Description

Fits a generalized additive model (GAM) to data using an approximate Bayesian inference technique based on penalized regression splines and Laplace approximations. Smooth additive terms are specified as a linear combination of a large number of cubic B-splines. To counterbalance the roughness of the fit, a discrete penalty on neighboring spline coefficients is imposed in the spirit of Eilers and Marx (1996). The effective degrees of freedom of the smooth terms are also estimated.

The optimal amount of smoothing is determined by a grid-based exploration of the posterior penalty space when the number of smooth terms is small to moderate. When the dimension of the penalty space is large, the optimal smoothing parameter is chosen to be the value that maximizes the (log-)posterior of the penalty vector. Approximate Bayesian credible intervals for latent model variables and functions of latent model variables are available.

Usage

```
gamlps(formula, data, K = 30, family = c("gaussian", "poisson", "bernoulli", "binomial"),
       gauss.scale, nbinom, penorder = 2, cred.int = 0.95)
```

Arguments

formula	A formula object where the ~ operator separates the response from the covariates of the linear part z1,z2,.. and the smooth terms. A smooth term is specified by using the notation sm(.). For instance, the formula y ~ z1+sm(x1)+sm(x2) specifies a generalized additive model of the form *g(mu) = b0+b1z1+f1(x1)+f2(x2)*, where *b0, b1* are the regression coefficients of the linear part and *f1(.)* and *f2(.)* are smooth functions of the continuous covariates *x1* and *x2* respectively. The function *g(.)* is the canonical link function.
data	Optional. A data frame to match the variables names provided in formula.
K	A positive integer specifying the number of cubic B-spline functions in the basis used to model the smooth terms. Default is K = 30 and allowed values are 15 <= K <= 60. The same basis dimension is used for each smooth term in the model. Also, the computational cost to fit the model increases with K.
family	The error distribution of the model. It is a character string that must partially match either "gaussian" for Normal data with an identity link, "poisson" for Poisson data with a log link, "bernoulli" or "binomial" for Bernoulli or Binomial data with a logit link.

gauss.scale	The scale parameter to be specified when family = "gaussian". It corresponds to the variance of the response.
nbinom	The number of experiments in the Binomial family.
penorder	The penalty order used on finite differences of the coefficients of contiguous B-splines. Can be either 2 for a second-order penalty (the default) or 3 for a third-order penalty.
cred.int	The level of the pointwise credible interval to be computed for the coefficients in the linear part of the model.

Details

The B-spline basis used to approximate a smooth additive component is computed with the function [cubicbs](). The lower (upper) bound of the B-spline basis is taken to be the minimum (maximum) value of the covariate associated to the smooth. For identifiability purposes, the B-spline matrices (computed over the observed covariates) are centered. The centering consists is subtracting from each column of the B-spline matrix, the corresponding column average of another B-spline matrix computed on a fine grid of equidistant values in the domain of the smooth term.

A hierarchical Gamma prior is imposed on the roughness penalty vector. A Newton-Raphson algorithm is used to compute the posterior mode of the (log-)posterior penalty vector. The latter algorithm uses analytically derived versions of the gradient and Hessian. When the number of smooth terms in the model is smaller or equal to 4, a grid-based strategy is used for posterior exploration of the penalty space. Above that threshold, the optimal amount of smoothness is determined by the posterior maximum value of the penalty vector. This strategy allows to keep the computational burden to fit the model relatively low and conserve good statistical performance.

Value

An object of class gamlps containing several components from the fit. Details can be found in [gamlps.object](). Details on the output printed by gamlps can be found in [print.gamlps](). Fitted smooth terms can be visualized with the [plot.gamlps]() routine.

Author(s)

Oswaldo Gressani <oswaldo_gressani@hotmail.fr>.

References

Hastie, T. J. and Tibshirani., RJ (1990). Generalized additive models. *Monographs on statistics and applied probability*, 43, 335.

Eilers, P.H.C. and Marx, B.D. (1996). Flexible smoothing with B-splines and penalties. *Statistical Science*, **11**(2): 89-121.

Gressani, O. and Lambert, P. (2018). Fast Bayesian inference using Laplace approximations in a flexible promotion time cure model based on P-splines. *Computational Statistical & Data Analysis* **124**: 151-167.

See Also

[cubicbs](), [gamlps.object](), [print.gamlps](), [plot.gamlps]()

Examples

```
set.seed(14)
sim <- simgamdata(n = 300, setting = 2, dist = "binomial", scale = 0.25)
dat <- sim$data
fit <- gamlps(y ~ z1 + z2 + sm(x1) + sm(x2), K = 15, data = dat,
              penorder = 2, family = "binomial", nbinom = 15)
fit

# Check fit and compare with target (in red)
opar <- par(no.readonly = TRUE)
par(mfrow = c(1, 2))
domx <- seq(-1, 1 ,length = 200)
plot(fit, smoo.index = 1, cred.int = 0.95, ylim = c(-2, 2))
lines(domx, sim$f[[1]](domx), type= "l", lty = 2, lwd = 2, col = "red")
plot(fit, smoo.index = 2, cred.int = 0.95, ylim = c(-3, 3))
lines(domx, sim$f[[2]](domx), type= "l", lty = 2, lwd = 2, col = "red")
par(opar)
```

---

gamlps.object   *Object resulting from the fit of a generalized additive model.*

Description

An object returned by the gamlps function consists in a list with various components related to the
fit of a generalized additive model with the Laplace-P-spline approach.

Value

A gamlps object has the following elements:

formula	The formula of the generalized additive model.
family	The chosen exponential family.
link	The link function used for the fit.
n	Sample size.
q	Total number of smooth terms.
K	Number of B-spline basis functions used for the fit.
penalty.order	Chosen penalty order.
latfield.dim	The dimension of the latent field. This is equal to the sum of the number of B-spline coefficients and the number of regression parameters related to the covariates in the linear part.
linear.coeff	Estimated linear regression coefficients. This is a matrix containing the posterior point estimate, standard deviation, z-score and lower/upper bounds of the credible interval.

spline.estim	The estimated B-spline coefficients. This is a list with q vectors of size K-1 representing the estimated B-spline amplitudes for each smooth term.
edf	Estimated effective degrees of freedom for each latent field variable.
Approx.signif	A matrix returning the observed test statistic and p-value for the approximate significance of smooth terms.
EDf	The estimated effective degrees of freedom of the smooth terms.
EDfHPD.95	95% HPD interval for the degrees of freedom of the smooth terms.
ED	The estimated degrees of freedom of the GAM model.
vmap	The maximum a posteriori of the (log) posterior penalty vector.
Cov.vmap	Covariance matrix of the (log) posterior penalty vector evaluated at vmap.
pen.family	The family of the posterior distribution for v. It is either "skew-normal" or "gaussian".
pendist.params	The parameterization for the posterior distribution of v. If the posterior of v belongs to the skew-normal family, then pendist.params is a matrix with as many rows as the number of smooth terms q. Each row contains the location, scale and shape parameter of the skew-normal distribution. If the posterior of v belongs to the Gaussian family, then pendist.params is a vector of length q, corresponding to vmap.
Covmaximum	The covariance matrix of the latent field evaluated at the posterior maximum value of the penalty vector.
latmaximum	The latent field value evaluated at the posterior maximum value of the penalty vector.
fitted.values	The fitted response values.
residuals	The response residuals.
r2.adj	The adjusted r-squared of the model indicating the proportion of the data variance explained by the model fit.
data	The data frame of the model.

Author(s)

Oswaldo Gressani <oswaldo_gressani@hotmail.fr>.

See Also

[gamlps](#), [print.gamlps](#), [plot.gamlps](#)

---

kidneytran   *Survival data of kidney transplant patients.*

Description

Survival data of kidney transplant patients from Section 1.7 of Klein and Moeschberger (2003).

Usage

```
data(kidneytran)
```

Format

A data frame with 863 rows and 6 columns.

obs Observation number.

time Time to death.

delta Event indicator, 1=Dead, 0=Alive.

gender Gender, 1=Male, 2=Female.

race Race, 1=White, 2=Black.

age Age in years.

Source

<https://cran.r-project.org/package=KMsurv>

References

Klein, J.P. and Moeschberger, M. L. (2003). Survival analysis: Techniques for Censored and Truncated Data (Second edition), Springer. ISBN 978-1-4419-2985-3

---

laryngeal   *Survival data of male laryngeal cancer patients.*

Description

Survival data of male patients with larynx cancer from Section 1.8 of Klein and Moeschberger (2003).

Usage

```
data(laryngeal)
```

Format

A data frame with 90 rows and 5 columns.

stage Stage of disease.

time Time to death in months.

age Age at diagnosis of larynx cancer.

diagyr Year of diagnosis of larynx cancer.

delta Event indicator, 1=Dead, 0=Alive.

Source

https://cran.r-project.org/package=KMsurv

References

Klein, J.P. and Moeschberger, M. L. (2003). Survival analysis: Techniques for Censored and Truncated Data (Second edition), Springer. ISBN 978-1-4419-2985-3

---

medicaid *Data from the 1986 Medicaid Consumer Survey.*

Description

Data from the 1986 Medicaid survey sponsored by the Health Care Financing Administration (USA). It can be used to illustrate generalized additive models with a log link for the number of doctor visits as a response variable. The dataset is studied in Gurmu (1997).

Usage

```
data(medicaid)
```

Format

A data frame with 485 rows and 10 columns.

numvisits Count of doctor office/clinic and health centre visits.

exposure Length of observation period for ambulatory care in days.

children Number of children in the household.

age Age of the respondent.

income1000 Annual household income in US dollars.

access Access to health services, 0=Low access, 100=High access.

pc1times1000 First principal component of three health status variables: functional limitations, acute conditions and chronic conditions.

maritalstat Marital status, 0=Other, 1=Married.

sex Gender, 1=Female, 0=Male.

race Race, 0=Other, 1=White.

Source

http://qed.econ.queensu.ca/jae/1997-v12.3/gurmu/

References

Gurmu, S.(1997). Semi-parametric estimation of hurdle regression models with an application to medicaid utilization. *Journal of Applied Econometrics* **12**(3):225-242.

---

melanoma	*Melanoma survival data.*

Description

Melanoma survival dataset with 205 patients suffering from skin cancer and operated for malignant melanoma at Odense University Hospital in Denmark.

Usage

```
data(melanoma)
```

Format

A data frame with 205 rows and 7 columns.

time Survival time in years.

status 1 Died from melanoma, 0 still alive or died from another event.

sex 1=Male, 0=Female.

age Age in years.

year Year of operation.

thickness Tumour thickness measured in mm.

ulcer 1=Presence of ulceration, 0=Absence of ulceration.

Source

http://www.stats.ox.ac.uk/pub/MASS4/

References

Venables W.N., and Ripley, B.D. (2002). Modern Applied Statistics with S, Fourth edition. Springer, New York. ISBN 0-387-95457-0.

Andersen, P.K., Borgan, O., Gill, R.D., and Keiding, N. (1993) Statistical Models based on Counting Processes. Springer.

---

penaltyplot   *Plot the approximate posterior distribution of the penalty vector.*

Description

The routine gives a graphical representation of the univariate approximate posterior distribution of the (log-)penalty parameters for objects of class *coxlps*, *curelps*, *amlps* and *gamlps*.

Usage

```
penaltyplot(object, dimension, ...)
```

Arguments

object   An object of class `coxlps`, `curelps`, `amlps` or `gamlps`.

dimension   For objects of class `amlps` and `gamlps`, the penalty vector can have a dimension larger than one, i.e. more than a single smooth term is present in the considered additive model. In that case, `dimension` is the penalty dimension to be plotted corresponding either to a scalar indicating the desired dimension or to a vector indicating more than one dimension. For instance, dimension = c(1,3) displays two separate plots of the (approximate) posterior distribution of the (log-)penalty parameter associated to the first and the third smooth function respectively.

...   Further arguments to be passed to the routine.

Details

When q, the number of smooth term in a (generalized) additive model is smaller than five, the exploration of the posterior penalty space is based on a grid strategy. In particular, the multivariate grid of dimension q is constructed by taking the Cartesian product of univariate grids in each dimension j = 1,...q. These univariate grids are obtained from a skew-normal fit to the conditional posterior p(vj|vmap[-j]),D), where vj is the (log-)penalty value associated to the jth smooth function and vmap[-j] is the posterior maximum of the (log-)penalty vector omitting the jth dimension. The routine displays the latter skew-normal distributions. When q>=5, inference is based on vmap and the grid is omitted to avoid computational overflow. In that case, the posterior distribution of the (log-)posterior penalty vector v is approximated by a multivariate Gaussian and the routine shows the marginal distributions.

Author(s)

Oswaldo Gressani <oswaldo_gressani@hotmail.fr>.

Examples

```
### Classic simulated data example (with simgamdata)

set.seed(123)
sim.data <- simgamdata(setting = 2, n = 250, dist = "gaussian", scale = 0.25)
```

```
plot(sim.data)          # Scatter plot of response
data <- sim.data$data   # Simulated data frame
# Fit model
fit <- amlps(y ~ z1 + z2 + sm(x1) + sm(x2), data = data, K = 15)
fit

# Penalty plot
opar <- par(no.readonly = TRUE)
par(mfrow = c(1, 2))
penaltyplot(fit, dimension = c(1, 2))
par(opar)
```

---

plot.amlps   *Plot smooth functions of an additive model object.*

Description

Displays a plot of the fitted additive smooth components of an `amlps` object. The routine can also be used to print a table of point and set estimates of an additive smooth term for a user-specified grid of values.

Usage

```
## S3 method for class 'amlps'
plot(x, xp, smoo.index, cred.int = 0.95, plot.cred = TRUE,
     np = 100, fit.col = "blue", shade.col = "gray75", show.plot = TRUE,
     show.info = TRUE, ...)
```

Arguments

x	An object of class `amlps`.
xp	A numeric vector of grid values on which to compute a point estimate and point-wise credible interval for the smooth function specified in `smoo.index`. The components of xp must be within the range of the observed covariate values for the corresponding smooth function. Results will be displayed in a table.
smoo.index	The index of the smooth function. For instance `smoo.index = 2` refers to the second smooth function specified in the `formula` of the `amlps` routine.
cred.int	The level of the pointwise credible interval to be computed for the smooth additive term. Default is `0.95`.
plot.cred	Logical. Should the credible intervals be plotted? Default is TRUE.
np	The number of points used to construct the plot of the smooth additive function. Default is 100 and allowed values are between 20 and 200.
fit.col	The color of the fitted curve.
shade.col	The shading color for the credible intervals.
show.plot	Logical. Should the plot be displayed? Default is TRUE.

show.info	Logical. Should the table of point and set estimates of the smooth function on the specified xp values be displayed? Default is TRUE.
...	Further arguments to be passed to plot.

Details

Produces a plot of a smooth additive term fitted with the amlps function. On the y-axis, the estimated effective dimension of the smooth term is also displayed. At the bottom of each plot, vertical ticks indicate the location of the covariate values. The labels on the x-axis correspond to the covariate name associated to the smooth term.

Value

If xp is unspecified (the default), the routine will only return a plot of the estimated smooth curve. Otherwise, it provides a list with the following components:

xp	The chosen points on which to compute the smooth fit.
sm.xp	The estimated smooth fit at points specified in xp.
sm.low	The lower bound of the pointwise credible interval for the smooth additive function at points specified in xp.
sm.up	The upper bound of the pointwise credible interval for the smooth additive function at points specified in xp.
cred.int	The chosen level to compute credible intervals.
smoo.index	The index of the smooth function.

Author(s)

Oswaldo Gressani <oswaldo_gressani@hotmail.fr>.

See Also

amlps, amlps.object, print.amlps

Examples

```
### Classic simulated data example

set.seed(3)

sim.data <- simgamdata(setting = 2, n = 200, dist = "gaussian", scale = 0.3)
plot(sim.data)        # Scatter plot of response
data <- sim.data$data  # Simulated data frame

# Fit model
fit <- amlps(y ~ z1 + z2 + sm(x1) + sm(x2), data = data, K = 20)
fit

# Plot fit of second function and results for a specific grid x
```

```
plot(fit, xp = c(-0.8, -0.4, 0, 0.4, 0.8), smoo.index = 2, ylim=c(-3, 3))
```

---

plot.coxlps   *Plot baseline hazard and survival curves from a coxlps object.*

Description

Produces a plot of the baseline hazard and/or survival based on a coxlps object.

Usage

```
## S3 method for class 'coxlps'
plot(x, S0 = TRUE, h0 = TRUE, cred.int = 0.95, overlay.km = FALSE,
     plot.cred = FALSE, np = 50, show.legend = TRUE, ...)
```

Arguments

x	An object of class coxlps.
S0	Logical. Should the estimated baseline survival be plotted?
h0	Logical. Should the estimated baseline hazard be plotted?
cred.int	The level for an approximate pointwise credible interval to be computed for the baseline curves. Default is 0.95.
overlay.km	A logical value indicating whether the Kaplan-Meier estimate should be plotted together with the smooth baseline survival curve. The default is FALSE.
plot.cred	Logical. Should the credible intervals be plotted ? Default is FALSE.
np	The number of points used to plot the smooth baseline functions. Default is 50 and allowed values are between 20 and 200.
show.legend	Logical. Should a legend be displayed?
...	Further arguments to be passed to plot.

Details

Plots for the baseline hazard and survival curves are computed on a grid (of length np) between 0 and the 99th percentile of follow-up times. When plot.cred is FALSE, the fit omits to compute the approximate pointwise credible intervals for plotting and hence is less computationally intensive. Vertical ticks on the x-axis correspond to the observed follow-up times.

Author(s)

Oswaldo Gressani <oswaldo_gressani@hotmail.fr>.

See Also

coxlps coxlps.object

Examples

```
## Simulate survival data
set.seed(6)
betas <- c(0.35, -0.20, 0.05, 0.80) # Regression coefficients
data <- simsurvdata(a = 1.8, b = 2, n = 200, betas = betas, censperc = 25)
simdat <- data$survdata

# Fit model
fit <- coxlps(Surv(time, delta) ~ x1 + x2 + x3 + x4, data = simdat)
plot(fit, h0 = FALSE, S0 = TRUE, overlay.km = FALSE, show.legend = FALSE)
domt <- seq(0, 5.5, length = 500)
lines(domt, data$S0(domt), type = "l", col = "red")
legend("topright", c("Bayesian LPS", "Target"), col = c("black", "red"),
       lty = c(1, 1), bty = "n", cex = 0.8)
```

---

plot.curelps	*Plot estimated survival functions and cure probability for the promotion time cure model.*

Description

The routine takes as input an object of class curelps and plots the estimated baseline survival curve, the population survival curve for a specific covariate profile and a a smooth curve for the cure probability. Approximate pointwise credible intervals are available.

Usage

```
## S3 method for class 'curelps'
plot(x, cred.int = 0.95, curvetype = c("baseline", "population",
     "probacure"), covar.profile, fit.col = "black", shade.col = "gray75",
      plot.cred = FALSE, np = 50, show.legend = TRUE, ...)
```

Arguments

x	An object of class curelps.
cred.int	The level for an approximate pointwise credible interval to be computed for the smooth curves. Default is 0.95.
curvetype	The curve to be plotted; baseline (the default) will plot the estimated baseline survival, population the population survival function for a profile of covariates given in covar.profile, and probacure the probability to be cured (for a profile of covariates given in covar.profile) given that the subject has survived until time t.

covar.profile   A numeric vector of the same length as the number of covariates in the model. This corresponds to the profile of covariates for which to compute the population survival function and cure probability estimates. The order of the covariates in covar.profile is the same as the order specified in formula of the curelps routine. Each component of covar.profile should be in the range of the observed values for the corresponding covariate. If covar.profile is left unspecified by the user, the default will be to take the median covariate values.

fit.col   The color used for the estimated survival curve.

shade.col   The color used for the shading of the credible intervals.

plot.cred   Logical. Should the credible intervals be plotted? Default is FALSE.

np   The number of points used to plot the smooth curves. Default is 50 and allowed values are between 20 and 200.

show.legend   Show the legend? Default is TRUE.

...   Further arguments to be passed to plot.

Details

When plot.cred is FALSE, the routine omits to compute the approximate pointwise credible intervals for plotting and hence is less computationally intensive.

Author(s)

Oswaldo Gressani <oswaldo_gressani@hotmail.fr>.

See Also

[curelps](#), [curelps.object](#), [curelps.extract](#), [print.curelps](#).

Examples

```
# Example on phase III clinical trial e1684 on melanoma data

data(ecog1684)

# Kaplan-Meier curve
plot(survfit(Surv(time, status) ~ 1, data = ecog1684), mark.time = TRUE)
fit <- curelps(Surv(time, status) ~ lt(age + trt + sex) +
               st(age + trt + sex), data = ecog1684, K = 20, penorder = 2)
fit
profile1 <- c(0, 1, 1, 0, 1, 1) # Mean age, trt = IFN, sex = Female.
profile2 <- c(0, 0, 1, 0, 0, 1) # Mean age, trt = control, sex = Female.
opar <- par(no.readonly = TRUE)
par(mfrow = c(1, 2))
plot(fit, curvetype = "probacure", plot.cred = TRUE, ylim = c(0,1),
     covar.profile = profile1, cred.int = 0.90,
     main = "Mean age, trt = IFN, sex = Female", cex.main = 0.8,
     show.legend = FALSE)
plot(fit, curvetype = "probacure", plot.cred = TRUE, ylim = c(0,1),
```

```
        covar.profile = profile2, cred.int = 0.90,
        main = "Mean age, trt = control, sex = Female", cex.main = 0.8,
        show.legend = FALSE)
  par(opar)
```

| plot.gamlps | *Plot smooth functions of a generalized additive model object.* |

Description

Displays a plot of the fitted additive smooth components of an `gamlps` object. The routine can also be used to print a table of point and set estimates of an additive smooth term for a user-specified grid of values.

Usage

```
## S3 method for class 'gamlps'
plot(x, xp, smoo.index, cred.int = 0.95, plot.cred = TRUE,
     np = 100, fit.col = "blue", shade.col = "gray75", show.plot = TRUE,
     show.info = TRUE, ...)
```

Arguments

x	An object of class `gamlps`.
xp	A numeric vector of grid values on which to compute a point estimate and point-wise credible interval for the smooth function specified in `smoo.index`. The components of `xp` must be within the range of the observed covariate values for the corresponding smooth function. Results will be displayed in a table.
smoo.index	The index of the smooth function. For instance `smoo.index = 2` refers to the second smooth function specified in the `formula` of the `amlps` routine.
cred.int	The level of the pointwise credible interval to be computed for the smooth additive term. Default is `0.95`.
plot.cred	Logical. Should the credible intervals be plotted? Default is `TRUE`.
np	The number of points used to construct the plot of the smooth additive function. Default is 100 and allowed values are between 20 and 200.
fit.col	The color of the fitted curve.
shade.col	The shading color for the credible intervals.
show.plot	Logical. Should the plot be displayed? Default is `TRUE`.
show.info	Logical. Should the table of point and set estimates of the smooth function on the specified `xp` values be displayed? Default is `TRUE`.
...	Further arguments to be passed to `plot`.

Details

Produces a plot of a smooth additive term fitted with the gamlps function. On the y-axis, the estimated effective dimension of the smooth term is also displayed. At the bottom of each plot, vertical ticks indicate the location of the covariate values. The labels on the x-axis correspond to the covariate name associated to the smooth term.

Value

If xp is unspecified (the default), the routine will only return a plot of the estimated smooth curve. Otherwise, it provides a list with the following components:

xp	The chosen points on which to compute the smooth fit.
sm.xp	The estimated smooth fit at points specified in xp.
sm.low	The lower bound of the pointwise credible interval for the smooth additive function at points specified in xp.
sm.up	The upper bound of the pointwise credible interval for the smooth additive function at points specified in xp.
cred.int	The chosen level to compute credible intervals.
smoo.index	The index of the smooth function.

Author(s)

Oswaldo Gressani <oswaldo_gressani@hotmail.fr>.

See Also

gamlps, gamlps.object, print.gamlps

---

print.amlps   *Print an additive partial linear model object.*

Description

Print an additive partial linear model object.

Usage

```
## S3 method for class 'amlps'
print(x, ...)
```

Arguments

x	An object of class amlps.
...	Additional arguments to be passed to print.

Details

Prints informative output of a fitted additive partial linear model. In particular, the model formula, sample size, number of B-splines in basis, number of smooth terms, the chosen penalty order, the latent field dimension, the estimated coefficients of the linear part, the estimated standard deviation of the error and the effective dimension of the smooth terms are displayed.

Author(s)

Oswaldo Gressani <oswaldo_gressani@hotmail.fr>.

See Also

[amlps](), [amlps.object](), [plot.amlps]()

---

print.coxlps   *Print a coxlps object.*

Description

Print method for a `coxlps` object.

Usage

```
## S3 method for class 'coxlps'
print(x, ...)
```

Arguments

x   An object of class `coxlps`.

...   Further arguments passed to print.

Details

Prints informative output of a fitted Cox proportional hazards model with Laplace-P-splines.

Author(s)

Oswaldo Gressani <oswaldo_gressani@hotmail.fr>.

See Also

[coxlps]()

---

print.curelps   *Print the fit of a promotion time cure model.*

Description

Print method for a `curelps` object.

Usage

```
## S3 method for class 'curelps'
print(x, ...)
```

Arguments

x	An object of class `curelps`.
...	Further arguments to be passed to print.

Details

Prints informative output of a fitted promotion time cure model with the Laplace-P-spline approach. In particular, the model formula, number of B-splines in basis, chosen penalty order, latent field dimension, sample size, number of events and effective model dimension are provided. The estimated model coefficients related to the cure probability (long-term survival) and the population hazard dynamics (short-term survival) are also provided, where `coef` is the point estimate, `sd.post` the posterior standard deviation, `z` is the Wald test statistic and `lower.95` and `upper.95` the lower, respectively upper bounds of the approximate 95% pointwise credible interval.

Author(s)

Oswaldo Gressani <oswaldo_gressani@hotmail.fr>.

See Also

[curelps](), [curelps.extract](), [plot.curelps]()

---

print.gamlps   *Print a generalized additive model object.*

Description

Print a generalized additive model object.

Usage

```
## S3 method for class 'gamlps'
print(x, ...)
```

Arguments

x	An object of class gamlps.
...	Further arguments to be passed to print.

Details

Prints informative output of a fitted generalized additive model. In particular, the model formula, sample size, number of B-splines in basis, number of smooth terms, the chosen penalty order, the latent field dimension, model degrees of freedom, the estimated coefficients of the linear part and the estimated effective degrees of freedom of the smooth terms are displayed.

Author(s)

Oswaldo Gressani <oswaldo_gressani@hotmail.fr>.

See Also

[gamlps](), [gamlps.object](), [plot.gamlps]()

---

simcuredata *Simulation of survival times for the promotion time cure model.*

Description

Generates right censored time-to-event data with a plateau in the Kaplan-Meier estimate.

Usage

```
simcuredata(n, censor = c("Uniform", "Weibull"), cure.setting = 1,
            info = TRUE, KapMeier = FALSE)
```

Arguments

n	Sample size.
censor	The censoring scheme. Either Uniform (the default) or Weibull.
cure.setting	A number indicating the desired cure percentage. If cure.setting = 1 (default) the cure percentage is around 20%. With cure.setting = 2 the cure percentage is around 30%.
info	Should information regarding the simulation setting be printed to the console? Default is TRUE.
KapMeier	Logical. Should the Kaplan-Meier curve of the generated data be plotted? Default is FALSE.

Details

Latent event times are generated following Bender et al. (2005), with a baseline distribution chosen to be a Weibull with mean 8 and variance 17.47. When `cure.setting = 1` the regression coefficients of the long-term survival part are chosen to yield a cure percentage around 20%, while `cure.setting = 2` yields a cure percentage around 30%. Censoring is either governed by a Uniform distribution on the support [20, 25] or by a Weibull distribution with shape parameter 3 and scale parameter 25.

Value

A list with the following components:

n	Sample size.
survdata	A data frame containing the simulated data.
beta.coeff	The regression coefficients pertaining to long-term survival.
gamma.coeff	The regression coefficients pertaining to short-term survival.
cure.perc	The cure percentage.
censor.perc	The percentage of censoring.
censor	The censoring scheme.
S0	The baseline survival function under the chosen Weibull parameterization.

Author(s)

Oswaldo Gressani <oswaldo_gressani@hotmail.fr>.

This function is based on a routine used to describe a simulation setting in Bremhorst and Lambert (2016). Special thanks go to Vincent Bremhorst who shared this routine during his PhD thesis.

References

Bender, R., Augustin, T. and Blettner, M. (2005). Generating survival times to simulate Cox proportional hazards models, *Statistics in Medicine* **24**(11): 1713-1723.

Bremhorst, V. and Lambert, P. (2016). Flexible estimation in cure survival models using Bayesian P-splines. *Computational Statistics & Data Analysis* **93**: 270-284.

Gressani, O. and Lambert, P. (2018). Fast Bayesian inference using Laplace approximations in a flexible promotion time cure model based on P-splines. *Computational Statistics & Data Analysis* **124**: 151-167.

Examples

```
set.seed(10)
sim <- simcuredata(n = 300, censor = "Weibull", KapMeier = TRUE)
```

---

simgamdata   *Simulation of data for (Generalized) additive models.*

Description

Simulation of a data set that can be used to illustrate the [amlps](#) or [gamlps](#) routines to fit (generalized) additive models with the Laplace-P-spline methodology.

Usage

```
simgamdata(setting = 1, n, dist = "gaussian", scale = 0.5, info = TRUE)
```

Arguments

setting	The simulation setting. The default is setting = 1 for a setting with three smooth terms, while setting = 2 is another setting with only two smooth terms. The coefficients of the linear part of the predictor are also different in the two settings.
n	The sample size to simulate.
dist	A character string to specify the response distribution. The default is "gaussian". Other distributions can be "poisson", "bernoulli" and "binomial".
scale	Used to tune the noise level for Gaussian and Poisson distributions.
info	Should information regarding the simulation be printed? Default is true.

Details

The simulation settings contain two covariates in the linear part of the predictor, namely $z1 \sim Bern(0.5)$ and $z2 \sim N(0,1)$. The smooth additive terms are inspired from Antoniadis et al. (2012). For Binomial data, the number of trials is fixed to 15.

Value

An object of class *simgam*. Plot of a *simgam* object yields a scatter plot of the generated response values.

data	A data frame.
f	The true smooth functions.
betas	The regression coefficients of the linear part. The first term is the intercept.
dist	The distribution of the response.

Author(s)

Oswaldo Gressani <oswaldo_gressani@hotmail.fr>.

References

Antoniadis, A., Gijbels, I., and Verhasselt, A. (2012). Variable selection in additive models using P-splines. *Technometrics* **54**(4): 425-438.

Examples

```
set.seed(10)
sim <- simgamdata(n = 150, dist = "poisson", scale = 0.3)
plot(sim)
```

---

simsurvdata   *Simulation of right censored survival times for the Cox model.*

Description

Generates right censored time-to-event data. Latent event times are drawn from a Weibull distribution, while censoring times are generated from an exponential distribution.

Usage

```
simsurvdata(a, b, n, betas, censperc, tmax = NULL)
```

Arguments

a, b	The shape parameter 'a>0' and scale parameter 'b>0' of the Weibull.
n	Sample size.
betas	A numeric vector of regression coefficients. Allowed components of 'betas' are in the interval *[-1 ,1]* and the total number of components cannot exceed 5.
censperc	A numeric value in *[0,100]* corresponding to the targeted percentage of censoring.
tmax	A maximum upper bound for the generated latent event times. Especially useful for a simulation study in which the observed event times are constrained to be generated in a fixed range.

Details

The Weibull baseline hazard is parameterized as follows (see Hamada et al. 2008 pp. 408-409) :

$$h_0(t) = (a/(b^a))t^{(a-1)}, t > 0.$$

The i*th* latent event time is denoted by *T_i* and is generated following Bender et al. (2005) as follows:

$$T_i = b(-log(U_i)exp(-\beta^T x_i))^{(1/a)},$$

where *U_i* is a uniform random variable obtained with 'runif(1)' , *x_i* is the i*th* row of a covariate matrix X of dimension 'c(n, length(betas))' where each component is generated from a standard Gaussian distribution and β is the vector of regression coefficients given by 'betas'.

Value

An object of class 'simsurvdata' which is a list with the following components:

sample.size	Sample size.
censoring	Censoring scheme. Either *No censoring* or *Exponential*.
num.events	Number of events.
censoring.percentage	
	The effective censoring percentage.
survdata	A data frame containing the simulated data.
regcoeffs	The true regression coefficients used to simulate the data.
S0	The baseline survival function under the chosen Weibull parameterization.
h0	The baseline hazard function under the chosen Weibull parameterization.
Weibull.mean	The mean of the Weibull used to generate latent event times.
Weibull.variance	
	The variance of the Weibull used to generate latent event times.

The 'print' method summarizes the generated right censored data and the 'plot' method produces a graph with time on the x axis and horizontal bars on the y axis corresponding either to an event or a right censored observation. If 'n > 25', only the 25 first observations are plotted.

Author(s)

Oswaldo Gressani <oswaldo_gressani@hotmail.fr>.

References

Bender, R., Augustin, T. and Blettner, M. (2005). Generating survival times to simulate Cox proportional hazards models, *Statistics in Medicine* **24**(11): 1713-1723.

Hamada, M. S., Wilson, A., Reese, C. S. and Martz, H. (2008). *Bayesian Reliability*. Springer Science and Business Media.

Examples

```
set.seed(10)
sim <- simsurvdata(a = 2, b = 1, n = 300, betas = c(0.8, -0.6), censperc = 25)
sim
plot(sim)
```

---

snmatch   *Fit a skew-normal distribution to a target density.*

Description

The routine fits a skew-normal univariate distribution to a target density. Parameters of the resulting skew-normal fit are estimated by the method of moments.

Usage

```
snmatch(x, y, p = c(0.025, 0.5, 0.975))
```

Arguments

x	A numeric vector on the domain of the target density.
y	The y-coordinates of the target density on grid x.
p	Vector of probabilities at which to compute quantiles of the skew-normal fit.

Details

The skew-normal density is parameterized by a location parameter μ, a scale parameter ω and a shape parameter ρ that regulates skewness. The probability density function at any x on the real line is:

$$p(x) = (2/\omega)\phi((x - \mu)/\omega)\psi(\rho(x - \mu)/\omega),$$

where $\phi()$ and $\psi()$ denote the standard Gaussian density and cumulative distribution function respectively (see Azzalini 2018). The first moment and second and third central moments of the target density are computed based on the x, y coordinates using the trapezoidal rule and matched against the theoretical moments of a skew-normal distribution. The solution to this system of equations is the method of moment estimate of the location, scale and shape parameters of a skew-normal density.

Value

A list with the following components:

location	Estimated location parameter.
scale	Estimated scale parameter.
shape	Estimated shape parameter.
snfit	Fitted values of the skew-normal density computed on an equally spaced grid between min(x) and max(x).
quant	Vector of quantiles of the skew-normal fit computed on the input vector of probabilities p.
xgrid	Equidistant grid on which the skew-normal fitted density is computed.

Author(s)

Oswaldo Gressani <oswaldo_gressani@hotmail.fr>.

References

Azzalini, A. (2018). The Skew-Normal and Related families. *Cambridge University Press*.

Examples

```
# Pdf of skew-normal density
sn.target <- function(x, location, scale, shape){
              val <- 2 * stats::dnorm(x, mean = location, sd = scale) *
              pnorm(shape * (x - location) / scale)
              return(val)
             }

# Extract x and y coordinates from target
x.grid <- seq(-2, 6, length = 200)
y.grid <- sapply(x.grid, sn.target, location = 0, scale = 2, shape = 3)

# Computation of the fit and graphical illustration
fit <- snmatch(x.grid, y.grid)
domx <- seq(-2, 6, length = 1000)
plot(domx, sapply(domx, sn.target, location = 0, scale = 2, shape = 3),
     type = "l", ylab = "f(x)", xlab = "x", lwd= 2)
lines(fit$xgrid, fit$snfit, type="l", col = "red", lwd = 2, lty = 2)
legend("topright", lty = c(1,2), col = c("black", "red"), lwd = c(2, 2),
       c("Target","SN fit"), bty="n")

# Extract estimated parameters
fit$location # Estimated location parameter
fit$scale    # Estimated scale parameter
fit$shape    # Estimated shape parameter
```

Index