Package ‘mable’

October 13, 2022

Title Maximum Approximate Bernstein/Beta Likelihood Estimation

Version 3.1.1

Date 2022-06-21

Author Zhong Guan [aut, cre]

Maintainer Zhong Guan `<zguan@iusb.edu>`

Depends R (>= 3.5.0)

Description Fit data from a continuous population with a smooth density on finite interval by an approximate Bernstein polynomial model which is a mixture of certain beta distributions and find maximum approximate Bernstein likelihood estimator of the unknown coefficients. Consequently, maximum likelihood estimates of the unknown density, distribution functions, and more can be obtained. If the support of the density is not the unit interval then transformation can be applied. This is an implementation of the methods proposed by the author of this package published in the Journal of Nonparametric Statistics: Guan (2016) <doi:10.1080/10485252.2016.1163349> and Guan (2017) <doi:10.1080/10485252.2017.1374384>. For d
variates, under some semiparametric regression models such as Cox proportional hazards model and the accelerated failure time model, the baseline survival function can be estimated smoothly based on general interval censored data.

License LGPL (>= 2.0, < 3)

LazyData true

Encoding UTF-8

Imports survival, graphics, stats, icenReg, parallel, doParallel, foreach, iterators, tcltk

Suggests mixtools, ICsurv, knitr, rmarkdown, pbapply, markdown, ks, multimode

BuildVignettes true

VignetteBuilder knitr, rmarkdown

RoxygenNote 7.2.0

NeedsCompilation yes

Repository CRAN

Date/Publication 2022-06-23 23:50:02 UTC

R topics documented:

`chicken.embryo` *Chicken Embryo Data*

Description

The chicken embryo dataset which contains `day`, number of days, and `nT`, the corresponding frequencies.

Usage

```
data(chicken.embryo)
```

Format

The format is: List of 2: day: int [1:21] 1 2 3 4 5 6 7 8 9 10 ...; nT : int [1:21] 6 5 11 2 2 3 0 0 0 0 ...

Source

Jassim, E. W., Grossman, M., Koops, W. J. And Luykx, R. A. J. (1996). Multi-phasic analysis of embryonic mortality in chickens. *Poultry Sci.* 75, 464-71.

References

Kuurman, W. W., Bailey, B. A., Koops, W. J. And Grossman, M. (2003). A model for failure of a chicken embryo to survive incubation. *Poultry Sci.* 82, 214-22.

Guan, Z. (2017) Bernstein polynomial model for grouped continuous data. *Journal of Nonparametric Statistics*, 29(4):831-848.

Examples

```
data(chicken.embryo)
```

`cosmesis` *Breast cosmesis data*

Description

Data contain the interval-censored times to cosmetic deterioration for breast cancer patients undergoing radiation or radiation plus chemotherapy.

Usage

```
data(cosmesis)
```

Format

A data frame with 94 observations on the following 3 variables.

- left left endpoint of the censoring interval in months
- right right endpoint of the censoring interval in months
- treat a factor with levels `RT` and `RCT` representing radiotherapy-only and radiation plus chemotherapy treatments, respectively

Source

Finkelstein, D. M. and Wolfe, R. A. (1985) A semiparametric model for regression analysis of interval-censored failure time data. Biometrics 41, 933–945.

References

Finkelstein, D. M. (1986) A proportional hazards model for interval-censored failure time data. Biometrics 42, 845–854.

Examples

```
data(cosmesis)
```

`dmixbeta` *Mixture Beta Distribution*

Description

Density, distribution function, quantile function and pseudorandom number generation for the Bernstein polynomial model, mixture of beta distributions, with shapes $(i+1, m-i+1)$, $i=0,\ldots,m$, given mixture proportions $p=(p_0,\ldots,p_m)$ and support `interval`.

Usage

```
dmixbeta(x, p, interval = c(0, 1))

pmixbeta(x, p, interval = c(0, 1))

qmixbeta(u, p, interval = c(0, 1))

rmixbeta(n, p, interval = c(0, 1))
```

Arguments

`x`	a vector of quantiles
`p`	a vector of `m+1` values. The `m+1` components of `p` must be nonnegative and sum to one for mixture beta distribution. See 'Details'.
`interval`	support/truncation interval `[a, b]`.
`u`	a vector of probabilities
`n`	sample size

Details

The density of the mixture beta distribution on an interval $[a,b]$ can be written as a Bernstein polynomial $f_m(x;p) = (b-a)^{-1}\sum_{i=0}^m p_i\beta_{mi}[(x-a)/(b-a)]/(b-a)$, where $p = (p_0,\ldots,p_m)$, $p_i \ge 0$, $\sum_{i=0}^m p_i = 1$ and $\beta_{mi}(u) = (m+1)\binom{m}{i}u^i(1-x)^{m-i}$, $i = 0,1,\ldots,m$, is the beta density with shapes $(i+1, m-i+1)$. The cumulative distribution function is $F_m(x;p) = \sum_{i=0}^m p_i B_{mi}[(x-a)/(b-a)]$, where $B_{mi}(u)$, $i = 0,1,\ldots,m$, is the beta cumulative distribution function with shapes $(i+1, m-i+1)$. If $\pi = \sum_{i=0}^m p_i < 1$, then f_m/π is a truncated desity on $[a,b]$ with cumulative distribution function F_m/π. The argument `p` may be any numeric vector of `m+1` values when `pmixbeta()` and and `qmixbeta()` return the integral function $F_m(x;p)$ and its inverse, respectively, and `dmixbeta()` returns a Bernstein polynomial $f_m(x;p)$. If components of `p` are not all nonnegative or do not sum to one, warning message will be returned.

Value

A vector of $f_m(x;p)$ or $F_m(x;p)$ values at x. `dmixbeta` returns the density, `pmixbeta` returns the cumulative distribution function, `qmixbeta` returns the quantile function, and `rmixbeta` generates pseudo random numbers.

Author(s)

Zhong Guan <zguan@iusb.edu>

References

Bernstein, S.N. (1912), Demonstration du theoreme de Weierstrass fondee sur le calcul des probabilities, Communications of the Kharkov Mathematical Society, 13, 1–2.

Guan, Z. (2016) Efficient and robust density estimation using Bernstein type polynomials. *Journal of Nonparametric Statistics*, 28(2):250-271.

Guan, Z. (2017) Bernstein polynomial model for grouped continuous data. *Journal of Nonparametric Statistics*, 29(4):831-848.

See Also

`mable`

Examples

```
# classical Bernstein polynomial approximation
a<--4; b<-4; m<-200
x<-seq(a,b,len=512)
u<-(0:m)/m
p<-dnorm(a+(b-a)*u)
plot(x, dnorm(x), type="l")
lines(x, (b-a)*dmixbeta(x, p, c(a, b))/(m+1), lty=2, col=2)
legend(a, dnorm(0), lty=1:2, col=1:2, c(expression(f(x)==phi(x)),
                 expression(B^{f}*(x))))
```

`dmixmvbeta` *Multivariate Mixture Beta Distribution*

Description

Density, distribution function, and pseudorandom number generation for the multivariate Bernstein polynomial model, mixture of multivariate beta distributions, with given mixture proportions $p = (p_0, \ldots, p_{K-1})$, given degrees $m = (m_1, \ldots, m_d)$, and support `interval`.

Usage

```
dmixmvbeta(x, p, m, interval = NULL)

pmixmvbeta(x, p, m, interval = NULL)

rmixmvbeta(n, p, m, interval = NULL)
```

Arguments

`x`	a matrix with `d` columns or a vector of length `d` within support hyperrectangle $[a,b] = [a_1,b_1] \times \cdots \times [a_d,b_d]$
`p`	a vector of `K` values. All components of `p` must be nonnegative and sum to one for the mixture multivariate beta distribution. See 'Details'.
`m`	a vector of degrees, $(m_1,\ldots,m_d)$
`interval`	a vector of two endpoints or a `d x 2` matrix, each row containing the endpoints of support/truncation interval for each marginal density. If missing, the i-th row is assigned as `c(min(x[,i]), max(x[,i]))`.
`n`	sample size

Details

`dmixmvbeta()` returns a linear combination f_m of d-variate beta densities on $[a,b]$, $\beta_{mj}(x) = \prod_{i=1}^d \beta_{m_i,j_i}[(x_i-a_i)/(b_i-a_i)]/(b_i-a_i)$, with coefficients $p(j_1,\ldots,j_d)$, $0 \le j_i \le m_i, i = 1,\ldots,d$, where $[a,b] = [a_1,b_1] \times \cdots \times [a_d,b_d]$ is a hyperrectangle, and the coefficients are arranged in the column-major order of $j = (j_1,\ldots,j_d)$, $p_0,\ldots,p_{K-1}$, where $K = \prod_{i=1}^d (m_i+1)$. `pmixmvbeta()` returns a linear combination F_m of the distribution functions of d-variate beta distribution.

If all p_i's are nonnegative and sum to one, then `p` are the mixture proportions of the mixture multivariate beta distribution.

`dtmixbeta` *Exponentially Tilted Mixture Beta Distribution*

Description

Density, distribution function, quantile function and pseudorandom number generation for the exponentially tilted mixture of beta distributions, with shapes $(i+1, m-i+1)$, $i = 0,\ldots,m$, given mixture proportions $p = (p_0,\ldots,p_m)$ and support `interval`.

Usage

```
dtmixbeta(x, p, alpha, interval = c(0, 1), regr, ...)

ptmixbeta(x, p, alpha, interval = c(0, 1), regr, ...)

qtmixbeta(u, p, alpha, interval = c(0, 1), regr, ...)

rtmixbeta(n, p, alpha, interval = c(0, 1), regr, ...)
```

Arguments

`x`	a vector of quantiles
`p`	a vector of `m+1` components of `p` must be nonnegative and sum to one for mixture beta distribution. See 'Details'.
`alpha`	regression coefficients
`interval`	support/truncation interval `[a, b]`.
`regr`	regressor vector function $r(x) = (1, r_1(x), ..., r_d(x))$ which returns n x (d+1) matrix, n=length(x)
`...`	additional arguments to be passed to regr
`u`	a vector of probabilities
`n`	sample size

Details

The density of the mixture exponentially tilted beta distribution on an interval $[a, b]$ can be written $f_m(x;p) = (b-a)^{-1}\exp(\alpha' r(x))\sum_{i=0}^m p_i\beta_{mi}[(x-a)/(b-a)]/(b-a)$, where $p = (p_0, \ldots, p_m)$, $p_i \ge 0$, $\sum_{i=0}^m p_i = 1$ and $\beta_{mi}(u) = (m+1)\binom{m}{i}u^i(1-x)^{m-i}$, $i = 0, 1, \ldots, m$, is the beta density with shapes $(i+1, m-i+1)$. The cumulative distribution function is $F_m(x;p) = \sum_{i=0}^m p_i B_{mi}[(x-a)/(b-a); alpha]$, where $B_{mi}(u; alpha)$, $i = 0, 1, \ldots, m$, is the exponentially tilted beta cumulative distribution function with shapes $(i+1, m-i+1)$.

Value

A vector of $f_m(x;p)$ or $F_m(x;p)$ values at x. `dmixbeta` returns the density, `pmixbeta` returns the cumulative distribution function, `qmixbeta` returns the quantile function, and `rmixbeta` generates pseudo random numbers.

Author(s)

Zhong Guan <zguan@iusb.edu>

References

Guan, Z., Application of Bernstein Polynomial Model to Density and ROC Estimation in a Semi-parametric Density Ratio Model

See Also

`mable`

Examples

```
# classical Bernstein polynomial approximation
a<--4; b<-4; m<-200
x<-seq(a,b,len=512)
u<-(0:m)/m
p<-dnorm(a+(b-a)*u)
plot(x, dnorm(x), type="l")
```

```
lines(x, (b-a)*dmixbeta(x, p, c(a, b))/(m+1), lty=2, col=2)
legend(a, dnorm(0), lty=1:2, col=1:2, c(expression(f(x)==phi(x)),
              expression(B^{f}*(x))))
```

`mable` *Mable fit of one-sample raw data with an optimal or given degree.*

Description

Maximum approximate Bernstein/Beta likelihood estimation based on one-sample raw data with an optimal selected by the change-point method among `m0:m1` or a preselected model degree `m`.

Usage

```
mable(
  x,
  M,
  interval = c(0, 1),
  IC = c("none", "aic", "hqic", "all"),
  vb = 0,
  controls = mable.ctrl(),
  progress = TRUE
)
```

Arguments

`x`	a (non-empty) numeric vector of data values.
`M`	a positive integer or a vector `(m0, m1)`. If `M = m` or `m0 = m1 = m`, then `m` is a preselected degree. If `m0<m1` it specifies the set of consective candidate model degrees `m0:m1` for searching an optimal degree, where `m1-m0>3`.
`interval`	a vector containing the endpoints of supporting/truncation interval `c(a,b)`
`IC`	information criterion(s) in addition to Bayesian information criterion (BIC). Current choices are "aic" (Akaike information criterion) and/or "qhic" (Hannan–Quinn information criterion).
`vb`	code for vanishing boundary constraints, -1: f0(a)=0 only, 1: f0(b)=0 only, 2: both, 0: none (default).
`controls`	Object of class `mable.ctrl()` specifying iteration limit and the convergence criterion eps. Default is `mable.ctrl`. See Details.
`progress`	if TRUE a text progressbar is displayed

Details

Any continuous density function f on a known closed supporting interval $[a,b]$ can be estimated by Bernstein polynomial $f_m(x;p)=\sum_{i=0}^m p_i\beta_{mi}[(x-a)/(b-a)]/(b-a)$, where $p=(p_0,\ldots,p_m)$, $p_i\ge 0$, $\sum_{i=0}^m p_i=1$ and $\beta_{mi}(u)=(m+1)\binom{m}{i}u^i(1-x)^{m-i}$, $i=0,1,\ldots,m$, is the beta density with shapes $(i+1,m-i+1)$. For each `m`, the MABLE of the coefficients `p`, the mixture proportions, are obtained using EM algorithm. The EM iteration for each candidate `m` stops if either the total absolute change of the log likelihood and the coefficients of Bernstein polynomial is smaller than `eps` or the maximum number of iterations `maxit` is reached.

If `m0<m1`, an optimal model degree is selected as the change-point of the increments of log-likelihood, log likelihood ratios, for $m\in\{m_0,m_0+1,\ldots,m_1\}$. Alternatively, one can choose an optimal degree based on the BIC (Schwarz, 1978) which are evaluated at $m\in\{m_0,m_0+1,\ldots,m_1\}$. The search for optimal degree `m` is stoped if either `m1` is reached with a warning or the test for change-point results in a p-value `pval` smaller than `sig.level`. The BIC for a given degree `m` is calculated as in Schwarz (1978) where the dimension of the model is $d=\#\{i:\hat{p}_i\ge\epsilon,i=0,\ldots,m\}-1$ and a default ϵ is chosen as `.Machine$double.eps`.

If data show a clearly multimodal distribution by plotting the histogram for example, the model degree is usually large. The range `M` should be large enough to cover the optimal degree and the computation is time-consuming. In this case the iterative method of moment with an initial selected by a method of mode which is implemented by `optimable` can be used to reduce the computation time.

Value

A list with components

- `m` the given or a selected degree by method of change-point
- `p` the estimated vector of mixture proportions $p=(p_0,\ldots,p_m)$ with the selected/given optimal degree `m`
- `mloglik` the maximum log-likelihood at degree `m`
- `interval` support/truncation interval `(a,b)`
- `convergence` An integer code. 0 indicates successful completion (all the EM iterations are convergent and an optimal degree is successfully selected in `M`). Possible error codes are
  - 1, indicates that the iteration limit `maxit` had been reached in at least one EM iteration;
  - 2, the search did not finish before `m1`.
- `delta` the convergence criterion `delta` value

and, if `m0<m1`,

- `M` the vector `(m0, m1)`, where `m1`, if greater than `m0`, is the largest candidate when the search stoped
- `lk` log-likelihoods evaluated at $m\in\{m_0,\ldots,m_1\}$
- `lr` likelihood ratios for change-points evaluated at $m\in\{m_0+1,\ldots,m_1\}$
- `ic` a list containing the selected information criterion(s)
- `pval` the p-values of the change-point tests for choosing optimal model degree
- `chpts` the change-points chosen with the given candidate model degrees

Note

Since the Bernstein polynomial model of degree m is nested in the model of degree $m+1$, the maximum likelihood is increasing in m. The change-point method is used to choose an optimal degree m. The degree can also be chosen by a method of moment and a method of mode which are implemented by function `optimal()`.

Author(s)

Zhong Guan <zguan@iusb.edu>

References

Guan, Z. (2016) Efficient and robust density estimation using Bernstein type polynomials. *Journal of Nonparametric Statistics*, 28(2):250-271.

See Also

`optimable`

Examples

```
# Vaal Rive Flow Data
 data(Vaal.Flow)
 x<-Vaal.Flow$Flow
 res<-mable(x, M = c(2,100), interval = c(0, 3000), controls =
        mable.ctrl(sig.level = 1e-8, maxit = 2000, eps = 1.0e-9))
 op<-par(mfrow = c(1,2),lwd = 2)
 layout(rbind(c(1, 2), c(3, 3)))
 plot(res, which = "likelihood", cex = .5)
 plot(res, which = c("change-point"), lgd.x = "topright")
 hist(x, prob = TRUE, xlim = c(0,3000), ylim = c(0,.0022), breaks = 100*(0:30),
  main = "Histogram and Densities of the Annual Flow of Vaal River",
  border = "dark grey",lwd = 1,xlab = "x", ylab = "f(x)", col  = "light grey")
 lines(density(x, bw = "nrd0", adjust = 1), lty = 4, col = 4)
 lines(y<-seq(0, 3000, length = 100), dlnorm(y, mean(log(x)),
                  sqrt(var(log(x)))), lty = 2, col = 2)
 plot(res, which = "density", add = TRUE)
 legend("top", lty = c(1, 2, 4), col = c(1, 2, 4), bty = "n",
 c(expression(paste("MABLE: ",hat(f)[B])),
        expression(paste("Log-Normal: ",hat(f)[P])),
               expression(paste("KDE: ",hat(f)[K]))))
 par(op)


# Old Faithful Data
 library(mixtools)
 x<-faithful$eruptions
 a<-0; b<-7
 v<-seq(a, b,len = 512)
 mu<-c(2,4.5); sig<-c(1,1)
```

```
pmix<-normalmixEM(x,.5, mu, sig)
lam<-pmix$lambda; mu<-pmix$mu; sig<-pmix$sigma
y1<-lam[1]*dnorm(v,mu[1], sig[1])+lam[2]*dnorm(v, mu[2], sig[2])
res<-mable(x, M = c(2,300), interval = c(a,b), controls  =
       mable.ctrl(sig.level = 1e-8, maxit = 2000L, eps = 1.0e-7))
op<-par(mfrow = c(1,2),lwd = 2)
layout(rbind(c(1, 2), c(3, 3)))
plot(res, which = "likelihood")
plot(res, which = "change-point")
hist(x, breaks = seq(0,7.5,len = 20), xlim = c(0,7), ylim = c(0,.7),
    prob  = TRUE,xlab = "t", ylab = "f(t)", col  = "light grey",
    main = "Histogram and Density of
              Duration of Eruptions of Old Faithful")
lines(density(x, bw = "nrd0", adjust = 1), lty = 4, col = 4, lwd = 2)
plot(res, which = "density", add = TRUE)
lines(v, y1, lty = 2, col = 2, lwd = 2)
legend("topright", lty = c(1,2,4), col = c(1,2,4), lwd = 2, bty = "n",
     c(expression(paste("MABLE: ",hat(f)[B](x))),
        expression(paste("Mixture: ",hat(f)[P](t))),
        expression(paste("KDE: ",hat(f)[K](t)))))
par(op)
```

`mable.aft` *Mable fit of Accelerated Failure Time Model*

Description

Maximum approximate Bernstein/Beta likelihood estimation for accelerated failure time model based on interval censored data.

Usage

```
mable.aft(
  formula,
  data,
  M,
  g = NULL,
  tau = NULL,
  x0 = NULL,
  controls = mable.ctrl(),
  progress = TRUE
)
```

Arguments

`formula`	regression formula. Response must be `cbind`. See 'Details'.
`data`	a dataset

`M`	a positive integer or a vector `(m0, m1)`. If `M = m` or `m0 = m1 = m`, then `m` is a preselected degree. If `m0 < m1` it specifies the set of consective candidate model degrees `m0:m1` for searching an optimal degree, where `m1-m0>3`.
`g`	the given d-vector of regression coefficients, default is zero vector.
`tau`	the right endpoint of the support or truncation interval $[0, \tau)$ of the baseline density. Default is `NULL` (unknown), otherwise if `tau` is given then it is taken as a known value of τ. See 'Details'.
`x0`	a working baseline covariate x_0, default is zero vector. See 'Details'.
`controls`	Object of class `mable.ctrl()` specifying iteration limit and other control options. Default is `mable.ctrl`.
`progress`	if `TRUE` a text progressbar is displayed

Details

Consider the accelerated failure time model with covariate for interval-censored failure time data: $S(t|x) = S(t\exp(\gamma'(x - x_0))|x_0)$, where x_0 is a baseline covariate. Let $f(t|x)$ and $F(t|x) = 1 - S(t|x)$ be the density and cumulative distribution functions of the event time given $X = x$, respectively. Then $f(t|x_0)$ on a truncation interval $[0, \tau]$ can be approximated by $f_m(t|x_0; p) = \tau^{-1}\sum_{i=0}^m p_i\beta_{mi}(t/\tau)$, where $p_i \ge 0$, $i = 0, \ldots, m$, $\sum_{i=0}^m p_i = 1$, $\beta_{mi}(u)$ is the beta denity with shapes $i + 1$ and $m - i + 1$, and τ is larger than the largest observed time, either uncensored time, or right endpoint of interval/left censored, or left endpoint of right censored time. So we can approximate $S(t|x_0)$ on $[0, \tau]$ by $S_m(t|x_0; p) = \sum_{i=0}^m p_i\bar{B}_{mi}(t/\tau)$, where $\bar{B}_{mi}(u)$ is the beta survival function with shapes $i + 1$ and $m - i + 1$.

Response variable should be of the form `cbind(l, u)`, where `(l,u)` is the interval containing the event time. Data is uncensored if `l = u`, right censored if `u = Inf` or `u = NA`, and left censored data if `l = 0`. The truncation time `tau` and the baseline `x0` should be chosen so that $S(t|x) = S(t\exp(\gamma'(x - x_0))|x_0)$ on $[\tau, \infty)$ is negligible for all the observed x.

The search for optimal degree `m` stops if either `m1` is reached or the test for change-point results in a p-value `pval` smaller than `sig.level`.

Value

A list with components

- `m` the given or selected optimal degree `m`
- `p` the estimate of `p = (p_0, ..., p_m)`, the coefficients of Bernstein polynomial of degree `m`
- `coefficients` the estimated regression coefficients of the AFT model
- `SE` the standard errors of the estimated regression coefficients
- `z` the z-scores of the estimated regression coefficients
- `mloglik` the maximum log-likelihood at an optimal degree `m`
- `tau.n` maximum observed time τ_n
- `tau` right endpoint of trucation interval $[0, \tau)$
- `x0` the working baseline covariates
- `egx0` the value of $e^{\gamma' x_0}$

- `convergence` an integer code, 1 indicates either the EM-like iteration for finding maximum likelihood reached the maximum iteration for at least one `m` or the search of an optimal degree using change-point method reached the maximum candidate degree, 2 indicates both occured, and 0 indicates a successful completion.
- `delta` the final `delta` if `m0` = `m1` or the final `pval` of the change-point for searching the optimal degree `m`;

and, if `m0<m1`,

- `M` the vector `(m0, m1)`, where `m1` is the last candidate when the search stoped
- `lk` log-likelihoods evaluated at $m \in \{m_0, \ldots, m_1\}$
- `lr` likelihood ratios for change-points evaluated at $m \in \{m_0 + 1, \ldots, m_1\}$
- `pval` the p-values of the change-point tests for choosing optimal model degree
- `chpts` the change-points chosen with the given candidate model degrees

Author(s)

Zhong Guan <zguan@iusb.edu>

References

Guan, Z. (2019) Maximum Approximate Likelihood Estimation in Accelerated Failure Time Model for Interval-Censored Data, arXiv:1911.07087.

See Also

`maple.aft`

Examples

```
## Breast Cosmesis Data
  bcos=cosmesis
  bcos2<-data.frame(bcos[,1:2], x=1*(bcos$treat=="RCT"))
  g <- 0.41 #Hanson and  Johnson 2004, JCGS
  aft.res<-mable.aft(cbind(left, right)~x, data=bcos2, M=c(1, 30), g=g, tau=100, x0=1)
  op<-par(mfrow=c(1,2), lwd=1.5)
  plot(x=aft.res, which="likelihood")
  plot(x=aft.res, y=data.frame(x=0), which="survival", model='aft', type="l", col=1,
      add=FALSE, main="Survival Function")
  plot(x=aft.res, y=data.frame(x=1), which="survival", model='aft', lty=2, col=1)
 legend("bottomleft", bty="n", lty=1:2, col=1, c("Radiation Only", "Radiation and Chemotherapy"))
  par(op)
```

`mable.ctrl` *Control parameters for mable fit*

Description

Control parameters for mable fit

Usage

```
mable.ctrl(
  sig.level = 0.01,
  eps = 1e-07,
  maxit = 5000L,
  eps.em = 1e-07,
  maxit.em = 5000L,
  eps.nt = 1e-07,
  maxit.nt = 1000L,
  tini = 1e-04
)
```

Arguments

`sig.level`	the sigificance level for change-point method of choosing optimal model degree
`eps`	convergence criterion for iteration involves EM like and Newton-Raphson iterations
`maxit`	maximum number of iterations involve EM like and Newton-Raphson iterations
`eps.em`	convergence criterion for EM like iteration
`maxit.em`	maximum number of EM like iterations
`eps.nt`	convergence criterion for Newton-Raphson iteration
`maxit.nt`	maximum number of Newton-Raphson iterations
`tini`	a small positive number used to make sure initial `p` is in the interior of the simplex

Value

a list of the arguments' values

Author(s)

Zhong Guan <zguan@iusb.edu>

`mable.decon` *Mable deconvolution with a known error density*

Description

Maximum approximate Bernstein/Beta likelihood estimation in additive density deconvolution model with a known error density.

Usage

```
mable.decon(
  y,
  gn = NULL,
  ...,
  M,
  interval = c(0, 1),
  IC = c("none", "aic", "hqic", "all"),
  vanished = TRUE,
 controls = mable.ctrl(maxit.em = 1e+05, eps.em = 1e-05, maxit.nt = 100, eps.nt =
    1e-10),
  progress = TRUE
)
```

Arguments

`y`	vector of observed data values
`gn`	error density function if known, default is NULL if unknown
`...`	additional arguments to be passed to gn
`M`	a vector `(m0, m1)` specifies the set of consective candidate model degrees, `M = m0:m1`. If gn is unknown then `M` a 2 x 2 matrix whose rows `(m0,m1)` and `(k0,k1)` specify lower and upper bounds for degrees `m` and `k`, respectively.
`interval`	a finite vector `(a,b)`, the endpoints of supporting/truncation interval if `gn` is known. Otherwise, it is a 2 x 2 matrix whose rows `(a,b)` and `(a1,b1)` specify supporting/truncation intervals of `X` and ϵ, respectively. See Details.
`IC`	information criterion(s) in addition to Bayesian information criterion (BIC). Current choices are "aic" (Akaike information criterion) and/or "qhic" (Hannan–Quinn information criterion).
`vanished`	logical whether the unknown error density vanishes at both end-points of `[a1,b1]`
`controls`	Object of class `mable.ctrl()` specifying iteration limit and other control options. Default is `mable.ctrl`.
`progress`	if `TRUE` a text progressbar is displayed

Details

Consider the additive measurement error model $Y = X + \epsilon$, where X has an unknown distribution F on a known support `[a,b]`, ϵ has a known or unknown distribution G, and X and ϵ are independent. We want to estimate density $f = F'$ based on independent observations, $y_i = x_i + \epsilon_i$, $i = 1, \ldots, n$, of Y. We approximate f by a Bernstein polynomial model on `[a,b]`. If $g = G'$ is unknown on a known support `[a1,b1]`, then we approximate g by a Bernstein polynomial model on `[a1,b1]`, $a1 < 0 < b1$. We assume $E(\epsilon) = 0$. AIC and BIC methods are used to select model degrees `(m,k)`.

Value

A `mable` class object with components, if g is known,

- `M` the vector `(m0, m1)`, where `m1` is the last candidate degree when the search stoped
- `m` the selected optimal degree `m`
- `p` the estimate of `p = (p_0, ..., p_m)`, the coefficients of Bernstein polynomial of degree `m`
- `lk` log-likelihoods evaluated at $m \in \{m_0, \ldots, m_1\}$
- `lr` likelihood ratios for change-points evaluated at $m \in \{m_0 + 1, \ldots, m_1\}$
- `convergence` An integer code. 0 indicates an optimal degree is successfully selected in `M`. 1 indicates that the search stoped at `m1`.
- `ic` a list containing the selected information criterion(s)
- `pval` the p-values of the change-point tests for choosing optimal model degree
- `chpts` the change-points chosen with the given candidate model degrees

if g is unknown,

- `M` the 2 x 2 matrix with rows `(m0, m1)` and `(k0,k1)`
- `nu_aic` the selected optimal degrees `(m,k)` using AIC method
- `p_aic` the estimate of `p = (p_0, ..., p_m)`, the coefficients of Bernstein polynomial model for f of degree `m` as in `nu_aic`
- `q_aic` the estimate of `q = (q_0, ..., q_k)`, the coefficients of Bernstein polynomial model for g of degree `k` as in `nu_aic`
- `nu_bic` the selected optimal degrees `(m,k)` using BIC method
- `p_bic` the estimate of `p = (p_0, ..., p_m)`, the coefficients of Bernstein polynomial model for f of degree `m` as in `nu_bic`
- `q_bic` the estimate of `q = (q_0, ..., q_k)`, the coefficients of Bernstein polynomial model for g of degree `k` as in `nu_bic`
- `lk` matrix of log-likelihoods evaluated at $m \in \{m_0, \ldots, m_1\}$ and $k \in \{k_0, \ldots, k_1\}$
- `aic` a matrix containing the Akaike information criterion(s) at $m \in \{m_0, \ldots, m_1\}$ and $k \in \{k_0, \ldots, k_1\}$
- `bic` a matrix containing the Bayesian information criterion(s) at $m \in \{m_0, \ldots, m_1\}$ and $k \in \{k_0, \ldots, k_1\}$

Author(s)

Zhong Guan <zguan@iusb.edu>

References

Guan, Z., (2019) Fast Nonparametric Maximum Likelihood Density Deconvolution Using Bernstein Polynomials, Statistica Sinica, doi:10.5705/ss.202018.0173

Examples

```
# A simulated normal dataset
set.seed(123)
mu<-1; sig<-2; a<-mu-sig*5; b<-mu+sig*5;
gn<-function(x) dnorm(x, 0, 1)
n<-50;
x<-rnorm(n, mu, sig); e<-rnorm(n); y<-x+e;
res<-mable.decon(y, gn, interval = c(a, b), M = c(5, 50))
op<-par(mfrow = c(2, 2),lwd = 2)
plot(res, which="likelihood")
plot(res, which="change-point", lgd.x="topright")
plot(xx<-seq(a, b, length=100), yy<-dnorm(xx, mu, sig), type="l", xlab="x",
    ylab="Density", ylim=c(0, max(yy)*1.1))
plot(res, which="density", types=c(2,3), colors=c(2,3))
# kernel density based on pure data
lines(density(x), lty=4, col=4)
legend("topright", bty="n", lty=1:4, col=1:4,
c(expression(f), expression(hat(f)[cp]), expression(hat(f)[bic]), expression(tilde(f)[K])))
plot(xx, yy<-pnorm(xx, mu, sig), type="l", xlab="x", ylab="Distribution Function")
plot(res, which="cumulative",  types=c(2,3), colors=c(2,3))
legend("bottomright", bty="n", lty=1:3, col=1:3,
    c(expression(F), expression(hat(F)[cp]), expression(hat(F)[bic])))
par(op)
```

`mable.dr` *MABLE in Desnity Ratio Model*

Description

Maximum approximate Bernstein/Beta likelihood estimation in a density ratio model based on two-sample raw data.

Usage

```
mable.dr(
  x,
  y,
  M,
  regr,
  ...,
  interval = c(0, 1),
  alpha = NULL,
```

```
  vb = 0,
  baseline = NULL,
  controls = mable.ctrl(),
  progress = TRUE,
  message = FALSE
)
```

Arguments

`x, y`	original two sample raw data, codex:"Control", y: "Case".
`M`	a positive integer or a vector `(m0, m1)`.
`regr`	regressor vector function $r(x) = (1, r_1(x), ..., r_d(x))$ which returns n x (d+1) matrix, n=length(x)
`...`	additional arguments to be passed to regr
`interval`	a vector `(a,b)` containing the endpoints of supporting/truncation interval of x and y.
`alpha`	initial regression coefficient, missing value is imputed by logistic regression
`vb`	code for vanishing boundary constraints, -1: f0(a)=0 only, 1: f0(b)=0 only, 2: both, 0: none (default).
`baseline`	the working baseline, "Control" or "Case", if `NULL` it is chosen to the one with smaller estimated lower bound for model degree.
`controls`	Object of class `mable.ctrl()` specifying iteration limit and the convergence criterion for EM and Newton iterations. Default is `mable.ctrl`. See Details.
`progress`	logical: should a text progressbar be displayed
`message`	logical: should warning messages be displayed

Details

Suppose that `x` ("control") and `y` ("case") are independent samples from f0 and f1 which samples satisfy f1(x)=f0(x)exp[alpha0+alpha'r(x)] with r(x)=(r1(x),...,r_d(x)). Maximum approximate Bernstein/Beta likelihood estimates of (alpha0,alpha), f0 and f1 are calculated. If support is (a,b) then replace r(x) by r[a+(b-a)x]. For a fixed `m`, using the Bernstein polynomial model for baseline f_0, MABLEs of f_0 and parameters alpha can be estimated by EM algorithm and Newton iteration. If estimated lower bound m_b for `m` based on `y` is smaller that that based on `x`, then switch `x` and `y` and f_1 is used as baseline. If `M=m` or `m0=m1=m`, then `m` is a preselected degree. If `m0<m1` it specifies the set of consective candidate model degrees `m0:m1` for searching an optimal degree by the change-point method, where `m1-m0>3`.

Value

A list with components

- `m` the given or a selected degree by method of change-point
- `p` the estimated vector of mixture proportions $p = (p_0, \dots, p_m)$ with the given or selected degree `m`
- `alpha` the estimated regression coefficients

- `mloglik` the maximum log-likelihood at degree `m`
- `interval` support/truncation interval `(a,b)`
- `baseline` ="control" if f_0 is used as baseline, or ="case" if f_1 is used as baseline.
- `M` the vector `(m0, m1)`, where `m1`, if greater than `m0`, is the largest candidate when the search stoped
- `lk` log-likelihoods evaluated at $m \in \{m_0, \ldots, m_1\}$
- `lr` likelihood ratios for change-points evaluated at $m \in \{m_0 + 1, \ldots, m_1\}$
- `pval` the p-values of the change-point tests for choosing optimal model degree
- `chpts` the change-points chosen with the given candidate model degrees

Author(s)

Zhong Guan <zguan@iusb.edu>

References

Guan, Z., Maximum Approximate Bernstein Likelihood Estimation of Densities in a Two-sample Semiparametric Model

Examples

```
# Hosmer and Lemeshow (1989):
# ages and the status of coronary disease (CHD) of 100 subjects
x<-c(20, 23, 24, 25, 26, 26, 28, 28, 29, 30, 30, 30, 30, 30, 32,
32, 33, 33, 34, 34, 34, 34, 35, 35, 36, 36, 37, 37, 38, 38, 39,
40, 41, 41, 42, 42, 42, 43, 43, 44, 44, 45, 46, 47, 47, 48, 49,
49, 50, 51, 52, 55, 57, 57, 58, 60, 64)
y<-c(25, 30, 34, 36, 37, 39, 40, 42, 43, 44, 44, 45, 46, 47, 48,
48, 49, 50, 52, 53, 53, 54, 55, 55, 56, 56, 56, 57, 57, 57, 57,
58, 58, 59, 59, 60, 61, 62, 62, 63, 64, 65, 69)
regr<-function(x) cbind(1,x)
chd.mable<-mable.dr(x, y, M=c(1, 15), regr, interval = c(20, 70))
chd.mable
```

`mable.dr.group` *Mable fit of the density ratio model based on grouped data*

Description

Maximum approximate Bernstein/Beta likelihood estimation in a density ratio model based on two-sample grouped data.

Usage

```
mable.dr.group(
  t,
  n0,
  n1,
  M,
  regr,
  ...,
  interval = c(0, 1),
  alpha = NULL,
  vb = 0,
  controls = mable.ctrl(),
  progress = TRUE,
  message = TRUE
)
```

Arguments

`t`	cutpoints of class intervals
`n0, n1`	frequencies of two sample data grouped by the classes specified by `t`. coden0:"Control", `n1`: "Case".
`M`	a positive integer or a vector `(m0, m1)`.
`regr`	regressor vector function $r(x) = (1, r_1(x), ..., r_d(x))$ which returns n x (d+1) matrix, n=length(x)
`...`	additional arguments to be passed to regr
`interval`	a vector `(a,b)` containing the endpoints of supporting/truncation interval of x and y.
`alpha`	a given regression coefficient, missing value is imputed by logistic regression
`vb`	code for vanishing boundary constraints, -1: f0(a)=0 only, 1: f0(b)=0 only, 2: both, 0: none (default).
`controls`	Object of class `mable.ctrl()` specifying iteration limit and the convergence criterion for EM and Newton iterations. Default is `mable.ctrl`. See Details.
`progress`	logical: should a text progressbar be displayed
`message`	logical: should warning messages be displayed

Details

Suppose that `n0` ("control") and `n1` ("case") are frequencies of independent samples grouped by the classes `t` from f0 and f1 which satisfy f1(x)=f0(x)exp[alpha0+alpha'r(x)] with r(x)=(r1(x),...,r_d(x)). Maximum approximate Bernstein/Beta likelihood estimates of (alpha0,alpha), f0 and f1 are calculated. If support is (a,b) then replace r(x) by r[a+(b-a)x]. For a fixed `m`, using the Bernstein polynomial model for baseline f_0, MABLEs of f_0 and parameters alpha can be estimated by EM algorithm and Newton iteration. If estimated lower bound m_b for `m` based on `n1` is smaller that that based on `n0`, then switch `n0` and `n1` and use f_1 as baseline. If `M=m` or `m0=m1=m`, then `m` is a preselected degree. If `m0<m1` it specifies the set of consective candidate model degrees `m0:m1` for searching an optimal degree by the change-point method, where `m1-m0>3`.

`mable.group` *Mable fit of one-sample grouped data by an optimal or a preselected model degree*

Description

Maximum approximate Bernstein/Beta likelihood estimation based on one-sample grouped data with an optimal selected by the change-point method among `m0:m1` or a preselected model degree `m`.

Usage

```
mable.group(
  x,
  breaks,
  M,
  interval = c(0, 1),
  IC = c("none", "aic", "hqic", "all"),
  vb = 0,
  controls = mable.ctrl(),
  progress = TRUE
)
```

Arguments

`x`	vector of frequencies
`breaks`	class interval end points
`M`	a positive integer or a vector `(m0, m1)`. If `M = m` or `m0 = m1 = m`, then `m` is a preselected degree. If `m0<m1` it specifies the set of consective candidate model degrees `m0:m1` for searching an optimal degree, where `m1-m0>3`.
`interval`	a vector containing the endpoints of support/truncation interval
`IC`	information criterion(s) in addition to Bayesian information criterion (BIC). Current choices are "aic" (Akaike information criterion) and/or "qhic" (Hannan–Quinn information criterion).
`vb`	code for vanishing boundary constraints, -1: f0(a)=0 only, 1: f0(b)=0 only, 2: both, 0: none (default).
`controls`	Object of class `mable.ctrl()` specifying iteration limit and the convergence criterion eps. Default is `mable.ctrl`. See Details.
`progress`	if TRUE a text progressbar is displayed

Details

Any continuous density function f on a known closed supporting interval $[a, b]$ can be estimated by Bernstein polynomial $f_m(x;p) = \sum_{i=0}^m p_i\beta_{mi}[(x-a)/(b-a)]/(b-a)$, where $p = (p_0, \dots, p_m)$, $p_i \ge 0$, $\sum_{i=0}^m p_i = 1$ and $\beta_{mi}(u) = (m+1)\binom{m}{i}u^i(1-x)^{m-i}$, $i = 0, 1, \dots, m$, is the beta density with shapes $(i+1, m-i+1)$. For each `m`, the MABLE of the coefficients `p`, the mixture proportions, are obtained using EM algorithm. The EM iteration for each candidate `m` stops if either the total absolute change of the log likelihood and the coefficients of Bernstein polynomial is smaller than `eps` or the maximum number of iterations `maxit` is reached.

If `m0<m1`, an optimal model degree is selected as the change-point of the increments of log-likelihood, log likelihood ratios, for $m \in \{m_0, m_0+1, \ldots, m_1\}$. Alternatively, one can choose an optimal degree based on the BIC (Schwarz, 1978) which are evaluated at $m \in \{m_0, m_0+1, \ldots, m_1\}$. The search for optimal degree `m` is stoped if either `m1` is reached with a warning or the test for change-point results in a p-value `pval` smaller than `sig.level`. The BIC for a given degree `m` is calculated as in Schwarz (1978) where the dimension of the model is $d = \#\{i : \hat{p}_i \geq \epsilon, i = 0, \ldots, m\} - 1$ and a default ϵ is chosen as `.Machine$double.eps`.

Value

A list with components

- `m` the given or a selected degree by method of change-point
- `p` the estimated `p` with degree `m`
- `mloglik` the maximum log-likelihood at degree `m`
- `interval` supporting interval `(a, b)`
- `convergence` An integer code. 0 indicates successful completion (all the EM iterations are convergent and an optimal degree is successfully selected in `M`). Possible error codes are
  - 1, indicates that the iteration limit `maxit` had been reached in at least one EM iteration;
  - 2, the search did not finish before `m1`.
- `delta` the convergence criterion `delta` value

and, if `m0<m1`,

- `M` the vector `(m0, m1)`, where `m1`, if greater than `m0`, is the largest candidate when the search stoped
- `lk` log-likelihoods evaluated at $m \in \{m_0, \ldots, m_1\}$
- `lr` likelihood ratios for change-points evaluated at $m \in \{m_0+1, \ldots, m_1\}$
- `ic` a list containing the selected information criterion(s)
- `pval` the p-values of the change-point tests for choosing optimal model degree
- `chpts` the change-points chosen with the given candidate model degrees

Author(s)

Zhong Guan <zguan@iusb.edu>

References

Guan, Z. (2017) Bernstein polynomial model for grouped continuous data. *Journal of Nonparametric Statistics*, 29(4):831-848.

See Also

mable.ic

Examples

```
## Chicken Embryo Data
 data(chicken.embryo)
 a<-0; b<-21
 day<-chicken.embryo$day
 nT<-chicken.embryo$nT
 Day<-rep(day,nT)
 res<-mable.group(x=nT, breaks=a:b, M=c(2,100), interval=c(a, b), IC="aic",
    controls=mable.ctrl(sig.level=1e-6,  maxit=2000, eps=1.0e-7))
 op<-par(mfrow=c(1,2), lwd=2)
 layout(rbind(c(1, 2), c(3, 3)))
 plot(res, which="likelihood")
 plot(res, which="change-point")
 fk<-density(x=rep((0:20)+.5, nT), bw="sj", n=101, from=a, to=b)
 hist(Day, breaks=seq(a,b,  length=12), freq=FALSE, col="grey",
          border="white", main="Histogram and Density Estimates")
 plot(res, which="density",types=1:2, colors=1:2)
 lines(fk, lty=2, col=2)
 legend("topright", lty=c(1:2), c("MABLE", "Kernel"), bty="n", col=c(1:2))
 par(op)
```

`mable.ic` *Mable fit based on one-sample interval censored data*

Description

Maximum approximate Bernstein/Beta likelihood estimation of density and cumulative/survival distributions functions based on interal censored event time data.

Usage

```
mable.ic(
  data,
  M,
  pi0 = NULL,
  tau = Inf,
  IC = c("none", "aic", "hqic", "all"),
  controls = mable.ctrl(),
  progress = TRUE
)
```

Arguments

`data`	a dataset either `data.frame` or an `n x 2` matrix.
`M`	an positive integer or a vector `(m0, m1)`. If `M = m` or `m0 = m1 = m`, then `m` is a preselected degree. If `m0 < m1` it specifies the set of consective candidate model degrees `m0:m1` for searching an optimal degree, where `m1-m0>3`.
`pi0`	Initial guess of $\pi = F(\tau_n)$. Without right censored data, `pi0 = 1`. See 'Details'.
`tau`	right endpoint of support $[0, \tau)$ must be greater than or equal to the maximum observed time
`IC`	information criterion(s) in addition to Bayesian information criterion (BIC). Current choices are "aic" (Akaike information criterion) and/or "qhic" (Hannan–Quinn information criterion).
`controls`	Object of class `mable.ctrl()` specifying iteration limit and other control options. Default is `mable.ctrl`.
`progress`	if `TRUE` a text progressbar is displayed

Details

Let $f(t)$ and $F(t) = 1 - S(t)$ be the density and cumulative distribution functions of the event time, respectively. Then $f(t)$ on $[0, \tau_n]$ can be approximated by $f_m(t; p) = \tau_n^{-1} \sum_{i=0}^{m} p_i \beta_{mi}(t/\tau_n)$, where $p_i \ge 0$, $i = 0, \ldots, m$, $\sum_{i=0}^{m} p_i = 1 - p_{m+1}$, $\beta_{mi}(u)$ is the beta denity with shapes $i + 1$ and $m - i + 1$, and τ_n is the largest observed time, either uncensored time, or right endpoint of interval/left censored, or left endpoint of right censored time. We can approximate $S(t)$ on $[0, \tau]$ by $S_m(t; p) = \sum_{i=0}^{m+1} p_i \bar{B}_{mi}(t/\tau)$, where $\bar{B}_{mi}(u)$, $i = 0, \ldots, m$, is the beta survival function with shapes $i + 1$ and $m - i + 1$, $\bar{B}_{m,m+1}(t) = 1$, $p_{m+1} = 1 - \pi$, and $\pi = F(\tau_n)$. For data without right-censored time, $p_{m+1} = 1 - \pi = 0$. The search for optimal degree `m` is stoped if either `m1` is reached or the test for change-point results in a p-value `pval` smaller than `sig.level`.

Each row of `data`, `(l, u)`, is the interval containing the event time. Data is uncensored if `l = u`, right censored if `u = Inf` or `u = NA`, and left censored data if `l = 0`.

Value

a class '`mable`' object with components

- `p` the estimated `p` with degree `m` selected by the change-point method
- `mloglik` the maximum log-likelihood at an optimal degree `m`
- `interval` support/truncation interval `(0, b)`
- `M` the vector `(m0,m1)`, where `m1` is the last candidate when the search stoped
- `m` the selected optimal degree by the method of change-point
- `lk` log-likelihoods evaluated at $m \in \{m_0, \ldots, m_1\}$
- `lr` likelihood ratios for change-points evaluated at $m \in \{m_0 + 1, \ldots, m_1\}$
- `tau.n` maximum observed time τ_n
- `tau` right endpoint of support $[0, \tau)$
- `ic` a list containing the selected information criterion(s)

- `pval` the p-values of the change-point tests for choosing optimal model degree
- `chpts` the change-points chosen with the given candidate model degrees
- `convergence` an integer code. 0 indicates successful completion(the iteration is convergent). 1 indicates that the maximum candidate degree had been reached in the calculation;
- `delta` the final `pval` of the change-point for selecting the optimal degree `m`;

Author(s)

Zhong Guan <zguan@iusb.edu>

References

Guan, Z. (2019) Maximum Approximate Bernstein Likelihood Estimation in Proportional Hazard Model for Interval-Censored Data, arXiv:1906.08882 .

See Also

`mable.group`

Examples

```
library(mable)
bcos=cosmesis
bc.res0<-mable.ic(bcos[bcos$treat=="RT",1:2], M=c(1,50), IC="none")
bc.res1<-mable.ic(bcos[bcos$treat=="RCT",1:2], M=c(1,50), IC="none")
op<-par(mfrow=c(2,2),lwd=2)
plot(bc.res0, which="change-point", lgd.x="right")
plot(bc.res1, which="change-point", lgd.x="right")
plot(bc.res0, which="survival", add=FALSE, xlab="Months", ylim=c(0,1), main="Radiation Only")
legend("topright", bty="n", lty=1:2, col=1:2, c(expression(hat(S)[CP]),
             expression(hat(S)[BIC])))
plot(bc.res1, which="survival", add=FALSE, xlab="Months", main="Radiation and Chemotherapy")
legend("topright", bty="n", lty=1:2, col=1:2, c(expression(hat(S)[CP]),
             expression(hat(S)[BIC])))
par(op)
```

`mable.mvar` *Maximum Approximate Bernstein Likelihood Estimate of Multivariate Density Function*

Description

Maximum Approximate Bernstein Likelihood Estimate of Multivariate Density Function

Usage

```
mable.mvar(
  x,
  M0 = 1,
  M,
  search = TRUE,
  interval = NULL,
  use.mar.deg = TRUE,
  high.dim = FALSE,
  criterion = c(”cdf”, ”pdf”),
  controls = mable.ctrl(),
  progress = TRUE
)
```

Arguments

`x`	an `n x d` matrix or `data.frame` of multivariate sample of size `n`
`M0`	a positive integer or a vector of `d` positive integers specify starting candidate degrees for searching optimal degrees.
`M`	a positive integer or a vector of `d` positive integers specify the maximum candidate or the given model degrees for the joint density.
`search`	logical, whether to search optimal degrees between `M0` and `M` or not but use `M` as the given model degrees for the joint density.
`interval`	a vector of two endpoints or a `d x 2` matrix, each row containing the endpoints of support/truncation interval for each marginal density. If missing, the i-th row is assigned as `c(min(x[,i]), max(x[,i]))`.
`use.mar.deg`	logical, if TRUE, the optimal degrees are selected based on marginal data, otherwise, the optimal degrees are those minimize the maximum L2 distance between marginal cdf or pdf estimated based on marginal data and the joint data. See details.
`high.dim`	logical, data are high dimensional/large sample or not if TRUE, run a slower version procedure which requires less memory
`criterion`	either cdf or pdf should be used for selecting optimal degrees. Default is "cdf"
`controls`	Object of class `mable.ctrl()` specifying iteration limit and the convergence criterion `eps`. Default is `mable.ctrl`. See Details.
`progress`	if TRUE a text progressbar is displayed

Details

A d-variate density f on a hyperrectangle $[a, b] = [a_1, b_1] \times \cdots \times [a_d, b_d]$ can be approximated by a mixture of d-variate beta densities on $[a, b]$, $\beta_{mj}(x) = \prod_{i=1}^{d} \beta_{m_i,j_i}[(x_i - a_i)/(b_i - a_i)]/(b_i - a_i)$, with proportion $p(j_1, \ldots, j_d)$, $0 \le j_i \le m_i, i = 1, \ldots, d$. Let $\tilde{F}_i$ $(\tilde{f}_i)$ be an estimate with degree $\tilde{m}_i$ of the i-th marginal cdf (pdf) based on marginal data `x[,i]`, $i = 1, \ldots, d$. If `search=TRUE` and `use.marginal=TRUE`, then the optimal degrees are $(\tilde{m}_1, \ldots, \tilde{m}_d)$. If `search=TRUE` and `use.marginal=FALSE`, then the optimal degrees $(\hat{m}_1, \ldots, \hat{m}_d)$ are those that minimize the maximum of L_2-distance between $\tilde{F}_i$ $(\tilde{f}_i)$ and the estimate of F_i (f_i) based on the joint data with degrees $m = (m_1, \ldots, m_d)$ for all m between M_0 and M if `criterion`="cdf" (`criterion`="pdf").

For large data and multimodal density, the search for the model degrees is very time-consuming. In this case, it is suggested that the degrees are selected based on marginal data using `mable` or `optimable`.

Value

A list with components

- `m` a vector of the selected optimal degrees by the method of change-point
- `p` a vector of the mixture proportions $p(j_1, \ldots, j_d)$, arranged in the column-major order of $j = (j_1, \ldots, j_d)$, $0 \le j_i \le m_i, i = 1, \ldots, d$.
- `mloglik` the maximum log-likelihood at an optimal degree `m`
- `pval` the p-values of change-points for choosing the optimal degrees for the marginal densities
- `M` the vector `(m1, m2, ... , md)`, where `mi` is the largest candidate degree when the search stoped for the `i`-th marginal density
- `interval` support hyperrectangle $[a, b] = [a_1, b_1] \times \cdots \times [a_d, b_d]$
- `convergence` An integer code. 0 indicates successful completion(the EM iteration is convergent). 1 indicates that the iteration limit `maxit` had been reached in the EM iteration;

Author(s)

Zhong Guan <zguan@iusb.edu>

References

Wang, T. and Guan, Z.,(2019) Bernstein Polynomial Model for Nonparametric Multivariate Density, *Statistics*, Vol. 53, no. 2, 321-338

See Also

`mable`, `optimable`

Examples

```
## Old Faithful Data

 a<-c(0, 40); b<-c(7, 110)
 ans<- mable.mvar(faithful, M = c(46,19), search =FALSE,
         interval = cbind(a,b), progress=FALSE)
 plot(ans, which="density")
 plot(ans, which="cumulative")
```

`mable.ph` *Mable fit of Cox's proportional hazards regression model*

Description

Maximum approximate Bernstein/Beta likelihood estimation in Cox's proportional hazards regression model based on interal censored event time data.

Usage

```
mable.ph(
  formula,
  data,
  M,
  g = NULL,
  pi0 = NULL,
  tau = Inf,
  x0 = NULL,
  controls = mable.ctrl(),
  progress = TRUE
)
```

Arguments

`formula`	regression formula. Response must be `cbind`. See 'Details'.
`data`	a dataset
`M`	a positive integer or a vector `(m0, m1)`. If `M = m` or `m0 = m1 = m`, then `m` is a preselected degree. If `m0<m1` it specifies the set of consective candidate model degrees `m0:m1` for searching an optimal degree, where `m1-m0>3`.
`g`	initial guess of d-vector of regression coefficients. See 'Details'.
`pi0`	Initial guess of $\pi(x_0) = F(\tau_n\vert x_0)$. Without right censored data, `pi0 = 1`. See 'Details'.
`tau`	right endpoint of support $[0,\tau)$ must be greater than or equal to the maximum observed time
`x0`	a working baseline covariate. See 'Details'.
`controls`	Object of class `mable.ctrl()` specifying iteration limit and other control options. Default is `mable.ctrl`.
`progress`	if `TRUE` a text progressbar is displayed

Details

Consider Cox's PH model with covariate for interval-censored failure time data: $S(t|x) = S(t|x_0)^{\exp(\gamma'(x-x_0))}$, where x_0 satisfies $\gamma'(x-x_0) \ge 0$. Let $f(t|x)$ and $F(t|x) = 1-S(t|x)$ be the density and cumulative distribution functions of the event time given $X = x$, respectively. Then $f(t|x_0)$ on $[0,\tau_n]$ can be approximated by $f_m(t|x_0,p) = \tau_n^{-1}\sum_{i=0}^m p_i\beta_{mi}(t/\tau_n)$, where $p_i \ge 0$, $i = 0,\ldots,m$, $\sum_{i=0}^m p_i =$

$1-p_{m+1}$, $\beta_{mi}(u)$ is the beta denity with shapes $i+1$ and $m-i+1$, and τ_n is the largest observed time, either uncensored time, or right endpoint of interval/left censored, or left endpoint of right censored time. So we can approximate $S(t|x_0)$ on $[0,\tau_n]$ by $S_m(t|x_0;p) = \sum_{i=0}^{m+1} p_i\bar B_{mi}(t/\tau_n)$, where $\bar B_{mi}(u)$, $i = 0,\ldots,m$, is the beta survival function with shapes $i+1$ and $m-i+1$, $\bar B_{m,m+1}(t) = 1$, $p_{m+1} = 1-\pi(x_0)$, and $\pi(x_0) = F(\tau_n|x_0)$. For data without right-censored time, $p_{m+1} = 1-\pi(x_0) = 0$.

Response variable should be of the form `cbind(l, u)`, where `(l, u)` is the interval containing the event time. Data is uncensored if `l = u`, right censored if `u = Inf` or `u = NA`, and left censored data if `l = 0`. The associated covariate contains d columns. The baseline `x0` should chosen so that $\gamma'(x-x_0)$ is nonnegative for all the observed x and all γ in a neighborhood of its true value.

A missing initial value of `g` is imputed by `ic_sp()` of package `icenReg`.

The search for optimal degree `m` stops if either `m1` is reached or the test for change-point results in a p-value `pval` smaller than `sig.level`. This process takes longer than `maple.ph` to select an optimal degree.

Value

A list with components

- `m` the selected/preselected optimal degree `m`
- `p` the estimate of $p = (p_0,\ldots,p_m,p_{m+1})$, the coefficients of Bernstein polynomial of degree `m`
- `coefficients` the estimated regression coefficients of the PH model
- `SE` the standard errors of the estimated regression coefficients
- `z` the z-scores of the estimated regression coefficients
- `mloglik` the maximum log-likelihood at an optimal degree `m`
- `tau.n` maximum observed time τ_n
- `tau` right endpoint of support $[0,\tau)$
- `x0` the working baseline covariates
- `egx0` the value of $e^{\gamma' x_0}$
- `convergence` an integer code, 1 indicates either the EM-like iteration for finding maximum likelihood reached the maximum iteration for at least one `m` or the search of an optimal degree using change-point method reached the maximum candidate degree, 2 indicates both occured, and 0 indicates a successful completion.
- `delta` the final `delta` if `m0 = m1` or the final `pval` of the change-point for searching the optimal degree `m`;

and, if `m0<m1`,

- `M` the vector `(m0, m1)`, where `m1` is the last candidate degree when the search stoped
- `lk` log-likelihoods evaluated at $m \in \{m_0,\ldots,m_1\}$
- `lr` likelihood ratios for change-points evaluated at $m \in \{m_0+1,\ldots,m_1\}$
- `pval` the p-values of the change-point tests for choosing optimal model degree
- `chpts` the change-points chosen with the given candidate model degrees

Author(s)

Zhong Guan <zguan@iusb.edu>

References

Guan, Z. Maximum Approximate Bernstein Likelihood Estimation in Proportional Hazard Model for Interval-Censored Data, Statistics in Medicine. 2020; 1–21. https://doi.org/10.1002/sim.8801.

See Also

`maple.ph`

Examples

```
# Ovarian Cancer Survival Data
require(survival)
futime2<-ovarian$futime
futime2[ovarian$fustat==0]<-Inf
ovarian2<-data.frame(age=ovarian$age, futime1=ovarian$futime,
     futime2=futime2)
ova<-mable.ph(cbind(futime1, futime2) ~ age, data = ovarian2,
     M=c(2,35), g=.16, x0=35)
op<-par(mfrow=c(2,2))
plot(ova, which = "likelihood")
plot(ova, which = "change-point")
plot(ova, y=data.frame(age=60), which="survival", add=FALSE, type="l",
      xlab="Days", main="Age = 60")
plot(ova, y=data.frame(age=65), which="survival", add=FALSE, type="l",
      xlab="Days", main="Age = 65")
par(op)
```

`mable.reg` *Mable fit of semiparametric regression model based on interval censored data*

Description

Wrapping all codemable fit of regression models in one function. Using maximum approximate Bernstein/Beta likelihood estimation to fit semiparametric regression models: Cox ph model, proportional odds(po) model, accelerated failure time model, and so on.

Usage

```
mable.reg(
  formula,
  data,
  model = c("ph", "aft"),
```

```
    M,
    g = NULL,
    pi0 = NULL,
    tau = Inf,
    x0 = NULL,
    eta = 1,
    controls = mable.ctrl(),
    progress = TRUE
  )
```

Arguments

`formula`	regression formula. Response must be of the form `cbind(l, u)`. See 'Details'.
`data`	a dataset
`model`	the model to fit. Current options are "ph" (Cox PH) or "`aft`" (accelerated failure time model)
`M`	a vector `(m0, m1)` specifies the set of consective integers as candidate degrees
`g`	an initial guess of the regression coefficients
`pi0`	Initial guess of $\pi(x_0) = F(\tau_n\vert x_0)$. Without right censored data, `pi0 = 1`. See 'Details'.
`tau`	right endpoint of support $[0, \tau)$ must be greater than or equal to the maximum observed time
`x0`	a working baseline covariate. See 'Details'.
`eta`	the given positive value of η. Used when `model="po"`.
`controls`	Object of class `mable.ctrl()` specifying iteration limit and other control options. Default is `mable.ctrl`.
`progress`	if `TRUE` a text progressbar is displayed

Details

For "ph" model a missing initial guess of the regression coefficients `g` is obtained by `ic_sp()` of package `icenReg`. For "`aft`" model a missing `g` is imputed by the rank estimate `aftsrr()` of package `aftgee` for right-censored data. For general interval censored observations, we keep the right-censored but replace the finite interval with its midpoint and fit the data by `aftsrr()` as a right-censored data.

Value

A 'mable_reg' class object

Author(s)

Zhong Guan <zguan@iusb.edu>

See Also

`mable.aft`, `mable.ph`

`maple.aft` *Mable fit of AFT model with given regression coefficients for AFT model*

Description

Maximum approximate profile likelihood estimation of Bernstein polynomial model in accelerated failure time based on interal censored event time data with given regression coefficients which are efficient estimates provided by other semiparametric methods.

Usage

```
maple.aft(
  formula,
  data,
  M,
  g,
  tau = NULL,
  x0 = NULL,
  controls = mable.ctrl(),
  progress = TRUE
)
```

Arguments

`formula`	regression formula. Response must be `cbind`. See 'Details'.
`data`	a dataset
`M`	a positive integer or a vector `(m0, m1)`. If `M = m` or `m0 = m1 = m`, then `m` is a preselected degree. If `m0 < m1` it specifies the set of consective candidate model degrees `m0:m1` for searching an optimal degree, where `m1-m0 > 3`.
`g`	the given d-vector of regression coefficients.
`tau`	the right endpoint of the support or truncation interval $[0,\tau)$ of the baseline density. Default is `NULL` (unknown), otherwise if `tau` is given then it is taken as a known value of τ. See 'Details'.
`x0`	a working baseline covariate x_0, default is zero vector. See 'Details'.
`controls`	Object of class `mable.ctrl()` specifying iteration limit and other control options. Default is `mable.ctrl`.
`progress`	if `TRUE` a text progressbar is displayed

Details

Consider the accelerated failure time model with covariate for interval-censored failure time data: $S(t|x) = S(t\exp(\gamma'(x-x_0))|x_0)$, where x_0 is a baseline covariate. Let $f(t|x)$ and $F(t|x) = 1-S(t|x)$ be the density and cumulative distribution functions of the event time given $X = x$, respectively. Then $f(t|x_0)$ on a support or truncation interval $[0,\tau]$ can be approximated by

$f_m(t|x_0;p) = \tau^{-1}\sum_{i=0}^m p_i\beta_{mi}(t/\tau)$, where $p_i \ge 0$, $i = 0,\ldots,m$, $\sum_{i=0}^m p_i = 1$, $\beta_{mi}(u)$ is the beta denity with shapes $i+1$ and $m-i+1$, and τ is larger than the largest observed time, either uncensored time, or right endpoint of interval/left censored, or left endpoint of right censored time. We can approximate $S(t|x_0)$ on $[0,\tau]$ by $S_m(t|x_0;p) = \sum_{i=0}^m p_i\bar{B}_{mi}(t/\tau)$, where $\bar{B}_{mi}(u)$ is the beta survival function with shapes $i+1$ and $m-i+1$.

Response variable should be of the form `cbind(l, u)`, where `(l,u)` is the interval containing the event time. Data is uncensored if `l = u`, right censored if `u = Inf` or `u = NA`, and left censored data if `l = 0`. The truncation time `tau` and the baseline `x0` should be chosen so that $S(t|x) = S(t\exp(\gamma'(x-x_0))|x_0)$ on $[\tau,\infty)$ is negligible for all the observed x.

The search for optimal degree `m` stops if either `m1` is reached or the test for change-point results in a p-value `pval` smaller than `sig.level`.

Value

A list with components

- `m` the selected optimal degree `m`
- `p` the estimate of $p = (p_0,\ldots,p_m)$, the coefficients of Bernstein polynomial of degree `m`
- `coefficients` the given regression coefficients of the AFT model
- `SE` the standard errors of the estimated regression coefficients
- `z` the z-scores of the estimated regression coefficients
- `mloglik` the maximum log-likelihood at an optimal degree `m`
- `tau.n` maximum observed time τ_n
- `tau` right endpoint of trucation interval $[0,\tau)$
- `x0` the working baseline covariates
- `egx0` the value of $e^{\gamma' x_0}$
- `convergence` an integer code, 1 indicates either the EM-like iteration for finding maximum likelihood reached the maximum iteration for at least one `m` or the search of an optimal degree using change-point method reached the maximum candidate degree, 2 indicates both occured, and 0 indicates a successful completion.
- `delta` the final `delta` if `m0 = m1` or the final `pval` of the change-point for searching the optimal degree `m`;

and, if `m0<m1`,

- `M` the vector `(m0, m1)`, where `m1` is the last candidate when the search stoped
- `lk` log-likelihoods evaluated at $m \in \{m_0,\ldots,m_1\}$
- `lr` likelihood ratios for change-points evaluated at $m \in \{m_0+1,\ldots,m_1\}$
- `pval` the p-values of the change-point tests for choosing optimal model degree
- `chpts` the change-points chosen with the given candidate model degrees

Author(s)

Zhong Guan <zguan@iusb.edu>

References

Guan, Z. (2019) Maximum Approximate Likelihood Estimation in Accelerated Failure Time Model for Interval-Censored Data, arXiv:1911.07087.

See Also

`mable.aft`

Examples

```
## Breast Cosmesis Data
  bcos=cosmesis
  bcos2<-data.frame(bcos[,1:2], x=1*(bcos$treat=="RCT"))
  g<-0.41 #Hanson and  Johnson 2004, JCGS,
  res1<-maple.aft(cbind(left, right)~x, data=bcos2, M=c(1,30),  g=g, tau=100, x0=1)
  op<-par(mfrow=c(1,2), lwd=1.5)
  plot(x=res1, which="likelihood")
  plot(x=res1, y=data.frame(x=0), which="survival", model='aft', type="l", col=1,
      add=FALSE, main="Survival Function")
  plot(x=res1, y=data.frame(x=1), which="survival", model='aft', lty=2, col=1)
 legend("bottomleft", bty="n", lty=1:2, col=1, c("Radiation Only", "Radiation and Chemotherapy"))
  par(op)
```

`maple.dr` *Maximum approximate profile likelihood estimate of the density ratio model*

Description

Select optimal degree with a given regression coefficients.

Usage

```
maple.dr(
  x,
  y,
  M,
  regr,
  ...,
  interval = c(0, 1),
  alpha = NULL,
  vb = 0,
  baseline = NULL,
  controls = mable.ctrl(),
  progress = TRUE,
  message = TRUE
)
```

Arguments

`x, y`	original two sample raw data, codex:"Control", y: "Case".
`M`	a positive integer or a vector `(m0, m1)`.
`regr`	regressor vector function $r(x) = (1, r_1(x), ..., r_d(x))$ which returns n x (d+1) matrix, n=length(x)
`...`	additional arguments to be passed to regr
`interval`	a vector `(a,b)` containing the endpoints of supporting/truncation interval of x and y.
`alpha`	a given regression coefficient, missing value is imputed by logistic regression
`vb`	code for vanishing boundary constraints, -1: f0(a)=0 only, 1: f0(b)=0 only, 2: both, 0: none (default).
`baseline`	the working baseline, "Control" or "Case", if `NULL` it is chosen to the one with smaller estimated lower bound for model degree.
`controls`	Object of class `mable.ctrl()` specifying iteration limit and the convergence criterion for EM and Newton iterations. Default is `mable.ctrl`. See Details.
`progress`	logical: should a text progressbar be displayed
`message`	logical: should warning messages be displayed

Details

Suppose that ("control") and `y` ("case") are independent samples from f0 and f1 which satisfy f1(x)=f0(x)exp[alpha0+alpha'r(x)] with r(x)=(r1(x),...,r_d(x)). Maximum approximate Bernstein/Beta likelihood estimates of f0 and f1 are calculated with a given regression coefficients which are efficient estimates provided by other semiparametric methods such as logistic regression. If support is (a,b) then replace r(x) by r[a+(b-a)x]. For a fixed `m`, using the Bernstein polynomial model for baseline f_0, MABLEs of f_0 and parameters alpha can be estimated by EM algorithm and Newton iteration. If estimated lower bound m_b for `m` based on `y` is smaller that that based on `x`, then switch `x` and `y` and f_1 is used as baseline. If `M=m` or `m0=m1=m`, then `m` is a preselected degree. If `m0<m1` it specifies the set of consective candidate model degrees `m0:m1` for searching an optimal degree by the change-point method, where `m1-m0>3`.

Value

A list with components

- `m` the given or a selected degree by method of change-point
- `p` the estimated vector of mixture proportions $p = (p_0, \ldots, p_m)$ with the given or selected degree `m`
- `alpha` the given regression coefficients
- `mloglik` the maximum log-likelihood at degree `m`
- `interval` support/truncation interval `(a,b)`
- `baseline` ="control" if f_0 is used as baseline, or ="case" if f_1 is used as baseline.
- `M` the vector `(m0, m1)`, where `m1`, if greater than `m0`, is the largest candidate when the search stoped

- `lk` log-likelihoods evaluated at $m \in \{m_0, \ldots, m_1\}$
- `lr` likelihood ratios for change-points evaluated at $m \in \{m_0 + 1, \ldots, m_1\}$
- `pval` the p-values of the change-point tests for choosing optimal model degree
- `chpts` the change-points chosen with the given candidate model degrees

Author(s)

Zhong Guan <zguan@iusb.edu>

References

Guan, Z., Maximum Approximate Bernstein Likelihood Estimation of Densities in a Two-sample Semiparametric Model

---

`maple.dr.group` *Maximum approximate profile likelihood estimate of the density ratio model for grouped data with given regression coefficients*

Description

Select optimal degree of Bernstein polynomial model for grouped data with a given regression coefficients.

Usage

```
maple.dr.group(
  t,
  n0,
  n1,
  M,
  regr,
  ...,
  interval = c(0, 1),
  alpha = NULL,
  vb = 0,
  controls = mable.ctrl(),
  progress = TRUE,
  message = TRUE
)
```

Arguments

`t`	cutpoints of class intervals
`n0, n1`	frequencies of two sample data grouped by the classes specified by `t`. coden0:"Control", `n1`: "Case".
`M`	a positive integer or a vector `(m0, m1)`.

`regr`	regressor vector function $r(x) = (1, r_1(x), ..., r_d(x))$ which returns n x (d+1) matrix, n=length(x)
`...`	additional arguments to be passed to regr
`interval`	a vector `(a,b)` containing the endpoints of supporting/truncation interval of x and y.
`alpha`	a given regression coefficient, missing value is imputed by logistic regression
`vb`	code for vanishing boundary constraints, -1: f0(a)=0 only, 1: f0(b)=0 only, 2: both, 0: none (default).
`controls`	Object of class `mable.ctrl()` specifying iteration limit and the convergence criterion for EM and Newton iterations. Default is `mable.ctrl`. See Details.
`progress`	logical: should a text progressbar be displayed
`message`	logical: should warning messages be displayed

Details

Suppose that `n0`("control") and `n1`("case") are frequencies of independent samples grouped by the classes `t` from f0 and f1 which satisfy f1(x)=f0(x)exp[alpha0+alpha'r(x)] with r(x)=(r1(x),...,r_d(x)). Maximum approximate Bernstein/Beta likelihood estimates of f0 and f1 are calculated with a given regression coefficients which are efficient estimates provided by other semiparametric methods such as logistic regression. If support is (a,b) then replace r(x) by r[a+(b-a)x]. For a fixed `m`, using the Bernstein polynomial model for baseline f_0, MABLEs of f_0 and parameters alpha can be estimated by EM algorithm and Newton iteration. If estimated lower bound m_b for `m` based on `n1` is smaller that that based on `n0`, then switch `n0` and `n1` and use f_1 as baseline. If `M=m` or `m0=m1=m`, then `m` is a preselected degree. If `m0<m1` it specifies the set of consective candidate model degrees `m0:m1` for searching an optimal degree by the change-point method, where `m1-m0>3`.

Value

A list with components

- `m` the given or a selected degree by method of change-point
- `p` the estimated vector of mixture proportions $p = (p_0, \ldots, p_m)$ with the given or selected degree `m`
- `alpha` the given regression coefficients
- `mloglik` the maximum log-likelihood at degree `m`
- `interval` support/truncation interval `(a,b)`
- `baseline` ="control" if f_0 is used as baseline, or ="case" if f_1 is used as baseline.
- `M` the vector `(m0, m1)`, where `m1`, if greater than `m0`, is the largest candidate when the search stoped
- `lk` log-likelihoods evaluated at $m \in \{m_0, \ldots, m_1\}$
- `lr` likelihood ratios for change-points evaluated at $m \in \{m_0 + 1, \ldots, m_1\}$
- `pval` the p-values of the change-point tests for choosing optimal model degree
- `chpts` the change-points chosen with the given candidate model degrees

Author(s)

Zhong Guan <zguan@iusb.edu>

References

Guan, Z., Application of Bernstein Polynomial Model to Density and ROC Estimation in a Semiparametric Density Ratio Model

---

`maple.ph` *Mable fit of the PH model with given regression coefficients*

Description

Maximum approximate profile likelihood estimation of Bernstein polynomial model in Cox's proportional hazards regression based on interal censored event time data with given regression coefficients which are efficient estimates provided by other semiparametric methods.

Usage

```
maple.ph(
  formula,
  data,
  M,
  g,
  pi0 = NULL,
  tau = Inf,
  x0 = NULL,
  controls = mable.ctrl(),
  progress = TRUE
)
```

Arguments

`formula`	regression formula. Response must be `cbind`. See 'Details'.
`data`	a dataset
`M`	a positive integer or a vector `(m0, m1)`. If `M = m` or `m0 = m1 = m`, then `m` is a preselected degree. If `m0 < m1` it specifies the set of consective candidate model degrees `m0:m1` for searching an optimal degree, where `m1-m0>3`.
`g`	the given d-vector of regression coefficients
`pi0`	Initial guess of $\pi(x_0) = F(\tau_n\|x_0)$. Without right censored data, `pi0 = 1`. See 'Details'.
`tau`	right endpoint of support $[0, \tau)$ must be greater than or equal to the maximum observed time
`x0`	a working baseline covariate. See 'Details'.
`controls`	Object of class `mable.ctrl()` specifying iteration limit and other control options. Default is `mable.ctrl`.
`progress`	if `TRUE` a text progressbar is displayed

Details

Consider Cox's PH model with covariate for interval-censored failure time data: $S(t|x) = S(t|x_0)^{\exp(\gamma'(x-x_0))}$, where x_0 satisfies $\gamma'(x-x_0) \geq 0$. Let $f(t|x)$ and $F(t|x) = 1-S(t|x)$ be the density and cumulative distribution functions of the event time given $X = x$, respectively. Then $f(t|x_0)$ on $[0, \tau_n]$ can be approximated by $f_m(t|x_0; p) = \tau_n^{-1} \sum_{i=0}^m p_i\beta_{mi}(t/\tau_n)$, where $p_i \geq 0$, $i = 0, \ldots, m$, $\sum_{i=0}^m p_i = 1 - p_{m+1}$, $\beta_{mi}(u)$ is the beta denity with shapes $i+1$ and $m-i+1$, and τ_n is the largest observed time, either uncensored time, or right endpoint of interval/left censored, or left endpoint of right censored time. So we can approximate $S(t|x_0)$ on $[0, \tau_n]$ by $S_m(t|x_0; p) = \sum_{i=0}^{m+1} p_i\bar{B}_{mi}(t/\tau_n)$, where $\bar{B}_{mi}(u)$, $i = 0, \ldots, m$, is the beta survival function with shapes $i+1$ and $m-i+1$, $\bar{B}_{m,m+1}(t) = 1$, $p_{m+1} = 1-\pi(x_0)$, and $\pi(x_0) = F(\tau_n|x_0)$. For data without right-censored time, $p_{m+1} = 1-\pi(x_0) = 0$.

Response variable should be of the form `cbind(l, u)`, where `(l, u)` is the interval containing the event time. Data is uncensored if `l = u`, right censored if `u = Inf` or `u = NA`, and left censored data if `l = 0`. The associated covariate contains d columns. The baseline `x0` should chosen so that $\gamma'(x-x_0)$ is nonnegative for all the observed x.

The search for optimal degree `m` stops if either `m1` is reached or the test for change-point results in a p-value `pval` smaller than `sig.level`.

Value

a class '`mable_reg`' object, a list with components

- `M` the vector `(m0, m1)`, where `m1` is the last candidate degree when the search stoped
- `m` the selected optimal degree `m`
- `p` the estimate of $p = (p_0, \ldots, p_m, p_{m+1})$, the coefficients of Bernstein polynomial of degree `m`
- `coefficients` the given regression coefficients of the PH model
- `mloglik` the maximum log-likelihood at an optimal degree `m`
- `lk` log-likelihoods evaluated at $m \in \{m_0, \ldots, m_1\}$
- `lr` likelihood ratios for change-points evaluated at $m \in \{m_0+1, \ldots, m_1\}$
- `tau.n` maximum observed time τ_n
- `tau` right endpoint of support $[0, \tau)$
- `x0` the working baseline covariates
- `egx0` the value of $e^{\gamma' x_0}$
- `convergence` an integer code. 0 indicates successful completion(the iteration is convergent). 1 indicates that the maximum candidate degree had been reached in the calculation;
- `delta` the final convergence criterion for EM iteration;
- `chpts` the change-points among the candidate degrees;
- `pom` the p-value of the selected optimal degree `m` as a change-point;

Author(s)

Zhong Guan <zguan@iusb.edu>

References

Guan, Z. (2019) Maximum Approximate Bernstein Likelihood Estimation in Proportional Hazard Model for Interval-Censored Data, arXiv:1906.08882 .

See Also

`mable.ph`

Examples

```
 ## Simulated Weibull data
   require(icenReg)
   set.seed(123)
   simdata<-simIC_weib(70, inspections = 5, inspectLength = 1)
   sp<-ic_sp(cbind(l, u) ~ x1 + x2, data = simdata)
   res0<-maple.ph(cbind(l, u) ~ x1 + x2, data = simdata, M=c(2,20),
        g=sp$coefficients, tau=7)
   op<-par(mfrow=c(1,2))
   plot(res0,  which=c("likelihood","change-point"))
   par(op)
   res1<-mable.ph(cbind(l, u) ~ x1 + x2, data = simdata, M=res0$m,
      g=c(.5,-.5), tau=7)
   op<-par(mfrow=c(1,2))
   plot(res1, y=data.frame(x=0, x2=0), which="density", add=FALSE, type="l",
       xlab="Time", main="Desnity Function")
   lines(xx<-seq(0, 7, len=512), dweibull(xx, 2,2), lty=2, col=2)
   legend("topright", bty="n", lty=1:2, col=1:2, c("Estimated","True"))
   plot(res1, y=data.frame(x=0, x2=0), which="survival", add=FALSE, type="l",
       xlab="Time", main="Survival Function")
   lines(xx, 1-pweibull(xx, 2, 2), lty=2, col=2)
   legend("topright", bty="n", lty=1:2, col=1:2, c("Estimated","True"))
   par(op)
```

`marginal.p` *The mixing proportions of marginal distribution from the mixture of multivariate beta distribution*

Description

The mixing proportions of marginal distribution from the mixture of multivariate beta distribution

Usage

```
marginal.p(p, m)
```

Arguments

`p`	the mixing proportions of the mixture of multivariate beta distribution
`m`	the model degrees `m=(m1,...,md)` of the mixture of multivariate beta distribution

Value

a list of mixing proportions of all the marginal distributions

---

`optim.gcp` *Choosing optimal model degree by gamma change-point method*

Description

Choose an optimal degree using gamma change-point model with two changing shape and scale parameters.

Usage

```
optim.gcp(obj)
```

Arguments

`obj`	a class "mable" or 'mable_reg' object containig a vector `M = (m0, m1)`, `lk`, log-likelihoods evaluated evaluated at $m \in \{m_0, \ldots, m_1\}$

Value

a list with components

- `m` the selected optimal degree `m`
- `M` the vector `(m0, m1)`, where `m1` is the last candidate when the search stoped
- `mloglik` the maximum log-likelihood at degree `m`
- `interval` support/truncation interval `(a, b)`
- `lk` log-likelihoods evaluated at $m \in \{m_0, \ldots, m_1\}$
- `lr` likelihood ratios for change-points evaluated at $m \in \{m_0 + 1, \ldots, m_1\}$
- `pval` the p-values of the change-point tests for choosing optimal model degree
- `chpts` the change-points chosen with the given candidate model degrees

Examples

```
# simulated data
p<-c(1:5,5:1)
p<-p/sum(p)
x<-rmixbeta(100, p)
res1<-mable(x, M=c(2, 50), IC="none")
m1<-res1$m[1]
res2<-optim.gcp(res1)
m2<-res2$m
op<-par(mfrow=c(1,2))
plot(res1, which="likelihood", add=FALSE)
plot(res2, which="likelihood")
#segments(m2, min(res1$lk), m2, res2$mloglik, col=4)
plot(res1, which="change-point", add=FALSE)
plot(res2, which="change-point")
par(op)
```

`optimable` *mable with degree selected by the method of moment and method of mode*

Description

Maximum Approximate Bernstein/Beta Likelihood Estimation with an optimal model degree estimated by the Method of Moment

Usage

```
optimable(
  x,
  interval,
  m = NULL,
  mu = NULL,
  lam = NULL,
  modes = NULL,
  nmod = 1,
  ushaped = FALSE,
  maxit = 50L
)
```

Arguments

`x`	a univariate sample data in `interval`
`interval`	a closed interval `c(a,b)`, default is [0,1]
`m`	initial degree, default is 2 times the number of modes `nmod`.

`mu`	a vector of component means of multimodal mixture density, default is NULL for unimodal or unknown
`lam`	a vector of mixture proportions of same length of `mu`
`modes`	a vector of the locations of modes, if it is NULL (default) and `multimode::locmodes()`
`nmod`	the number of modes, if `nmod`=0, the lower bound for m is estimated based on mean and variance only.
`ushaped`	logical, whether or not the density is clearly U-shaped including J- and L-shaped with mode occurs at the endpoint of the support.
`maxit`	maximum iterations

Details

If the data show a clear uni- or multi-modal distribution, then give the value of `nmod` as the number of modes. Otherwise `nmod`=0. The degree is estimated by the iterative method of moment with an initial degree estimated by the method of mode. For multimodal density, if useful estimates of the component means `mu` and proportions `lam` are available then they can be used to give an initial degree. If the distribution is clearly U-, J-, or L-shaped, i.e., the mode occurs at the endpoint of `interval`, then set `ushaped`=TRUE. In this case the degree is estimated by the method of mode.

Value

A class "mable" object with components

- `m` the given or a selected degree by method of change-point
- `p` the estimated vector of mixture proportions $p = (p_0, \ldots, p_m)$ with the selected/given optimal degree `m`
- `mloglik` the maximum log-likelihood at degree `m`
- `interval` support/truncation interval `(a,b)`
- `convergence` An integer code. 0 indicates successful completion (all the EM iterations are convergent and an optimal degree is successfully selected in `M`). Possible error codes are
  - 1, indicates that the iteration limit `maxit` had been reached in at least one EM iteration;
  - 2, the search did not finish before `m1`.
- `delta` the convergence criterion `delta` value

Author(s)

Zhong Guan <zguan@iusb.edu>

Examples

```
## Old Faithful Data
x<-faithful
x1<-faithful[,1]
x2<-faithful[,2]
a<-c(0, 40); b<-c(7, 110)
mu<-(apply(x,2,mean)-a)/(b-a)
```

```
s2<-apply(x,2,var)/(b-a)^2
# mixing proportions
lambda<-c(mean(x1<3),mean(x2<65))
# guess component mean
mu1<-(c(mean(x1[x1<3]), mean(x2[x2<65]))-a)/(b-a)
mu2<-(c(mean(x1[x1>=3]), mean(x2[x2>=65]))-a)/(b-a)
# estimate lower bound for m
mb<-ceiling((mu*(1-mu)-s2)/(s2-lambda*(1-lambda)*(mu1-mu2)^2)-2)
mb
m1<-optimable(x1, interval=c(a[1],b[1]), nmod=2, modes=c(2,4.5))$m
m2<-optimable(x2, interval=c(a[2],b[2]), nmod=2, modes=c(52.5,80))$m
m1;m2
erupt1<-mable(x1, M=mb[1], interval=c(a[1],b[1]))
erupt2<-mable(x1, M=m1, interval=c(a[1],b[1]))
wait1<-mable(x2, M=mb[2],interval=c(a[2],b[2]))
wait2<-mable(x2, M=m2,interval=c(a[2],b[2]))
ans1<- mable.mvar(faithful, M = mb, search =FALSE, interval = cbind(a,b))
ans2<- mable.mvar(faithful, M = c(m1,m2), search =FALSE, interval = cbind(a,b))
op<-par(mfrow=c(1,2), cex=0.8)
hist(x1, probability = TRUE, col="grey", border="white", main="",
      xlab="Eruptions", ylim=c(0,.65), las=1)
plot(erupt1, add=TRUE,"density")
plot(erupt2, add=TRUE,"density",lty=2,col=2)
legend("topleft", lty=c(1,2),col=1:2, bty="n", cex=.7,
      c(expression(paste("m = ", m[b])),expression(paste("m = ", hat(m)))))
hist(x2, probability = TRUE, col="grey", border="white", main="",
      xlab="Waiting", las=1)
plot(wait1, add=TRUE,"density")
plot(wait2, add=TRUE,"density",lty=2,col=2)
legend("topleft", lty=c(1,2),col=1:2, bty="n", cex=.7,
      c(expression(paste("m = ", m[b])),expression(paste("m = ", hat(m)))))
par(op)
op<-par(mfrow=c(1,2), cex=0.7)
plot(ans1, which="density", contour=TRUE)
plot(ans2, which="density", contour=TRUE, add=TRUE, lty=2, col=2)
plot(ans1, which="cumulative", contour=TRUE)
plot(ans2, which="cumulative", contour=TRUE, add=TRUE, lty=2, col=2)
par(op)
```

`pancreas` *Pancreatic Cancer Biomarker Data*

Description

Contain sera measurements from 51 control patients with pancreatitis and 90 case patients with pancreatic cancer at the Mayo Clinic with a cancer antigen, CA125, and with a carbohydrate antigen, CA19-9 (Wieand, et al, 1989)

Usage

```
data(pancreas)
```

Format

A data frame with 141 rows and 3 variables.

- ca199. CA19-9 levels
- ca125. CA125 levels
- status. 0 = controls (non-cancer) and 1 = cases (cancer).

Source

Wieand, S., Gail, M. H., James, B. R., and James, K.L. (1989). A family of nonparametric statistics for comparing diagnostic markers with paired or unpaired data. Biometrika, 76, 585–592.

References

Wieand, S., Gail, M. H., James, B. R., and James, K.L. (1989). A family of nonparametric statistics for comparing diagnostic markers with paired or unpaired data. Biometrika, 76, 585–592.

Examples

```
data(pancreas)
```

`plot.mable` *Plot mathod for class 'mable'*

Description

Plot mathod for class 'mable'

Usage

```
## S3 method for class 'mable'
plot(
  x,
 which = c("density", "cumulative", "survival", "likelihood", "change-point", "all"),
  add = FALSE,
  contour = FALSE,
  lgd.x = NULL,
  lgd.y = NULL,
  nx = 512,
  ...
)
```

Arguments

`x`	Class "mable" object return by `mablem`, `mable`, `mablem.group` or `mable.group` functions which contains `p`, `mloglik`, and `M` = `m0:m1`, `lk`, `lr`,
`which`	indicates which graphs to plot, options are "density", "cumulative", "likelihood", "change-point", "all". If not "all", `which` can contain more than one options.
`add`	logical add to an existing plot or not
`contour`	logical plot contour or not for two-dimensional data
`lgd.x, lgd.y`	coordinates of position where the legend is displayed
`nx`	number of evaluations of density, or cumulative distribution curve to be plotted.
`...`	additional arguments to be passed to the base plot function

Value

The data used for 'plot()', 'lines()', or 'persp()' are returned invisibly.

`plot.mable_reg` *Plot mathod for class 'mable_reg'*

Description

Plot mathod for class 'mable_reg'

Usage

```
## S3 method for class 'mable_reg'
plot(
  x,
  y,
  newdata = NULL,
  ntime = 512,
  xlab = "Time",
  which = c("survival", "likelihood", "change-point", "density", "all"),
  add = FALSE,
  ...
)
```

Arguments

`x`	a class 'mable_reg' object return by functions such as `mable.ph` which contains `M`, `coefficients`, `p`, `m`, `x0`, `tau.n`, `tau` `lk`, `lr`.
`y`	a new data.frame of covariate value(s) as row(s), whose columns are arranged in the same order as in the `formula` called by the function that returned the object `x`.
`newdata`	a new data.frame (ignored if `y` is included), imputed by the working baseline `x0` if both missing.

`ntime`	number of evaluations of density, survival or cumulative distribution curve to be plotted.
`xlab`	x-axis label
`which`	indicates which graphs to plot, options are "survival", "likelihood", "change-point", "density", or "all". If not "all", `which` can contain more than one options.
`add`	logical add to an existing plot or not
`...`	additional arguments to be passed to the base plot function

Author(s)

Zhong Guan <zguan@iusb.edu>

`se.coef.dr` *Standard errors of coefficients in density ratio model*

Description

Bootstrap estimates of standard errors for the regression coefficients which are estimated by maximum approximate Bernstein/Beta likelihood estimation method in a density ratio model based on two-sample raw data.

Usage

```
se.coef.dr(
  obj,
  grouped = FALSE,
  B = 500,
  parallel = FALSE,
  ncore = NULL,
  controls = mable.ctrl()
)
```

Arguments

`obj`	Class 'mable_dr' object return by `mable.dr` or `mable.dr.group` functions
`grouped`	logical: are data grouped or not.
`B`	number of bootstrap runs.
`parallel`	logical: do parallel or not.
`ncore`	number of cores used for parallel computing. Default is half of availables.
`controls`	Object of class `mable.ctrl()` specifying iteration limit and the convergence criterion for EM and Newton iterations. Default is `mable.ctrl`. See Details.

Details

Bootstrap method is used based on bootstrap samples generated from the MABLE's of the densities f0 and f1. The bootstrap samples are fitted by the Bernstein polynomial model and the `glm()` to obtain bootstrap versions of coefficient estimates.

Value

the estimated standard errors

`summary.mable` *Summary mathods for classes 'mable' and 'mable_reg'*

Description

Produces a summary of a mable fit.

Usage

```
## S3 method for class 'mable'
summary(object, ...)

## S3 method for class 'mable_reg'
summary(object, ...)
```

Arguments

`object`	Class "mable" or 'mable_reg' object return by `mable` or `mable.xxxx` functions
`...`	for future methods

Value

Invisibly returns its argument, `object`.

Examples

```
## Breast Cosmesis Data
  bcos=cosmesis
  bcos2<-data.frame(bcos[,1:2], x=1*(bcos$treat=="RCT"))
  aft.res<-mable.aft(cbind(left, right)~x, data=bcos2, M=c(1, 30), g=.41,
       tau=100, x0=1)
  summary(aft.res)
```

`Vaal.Flow` *Vaal River Annual Flow Data*

Description

The annual flow data of Vaal River at Standerton as given by Table 1.1 of Linhart and Zucchini (1986) give the flow in millions of cubic metres.

Usage

```
data(Vaal.Flow)
```

Format

The format is: int [1:65] 222 1094 452 1298 882 988 276 216 103 490 ...

References

Linhart, H., and Zucchini, W., *Model Selection*, Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics, New York: John Wiley and Sons Inc, 1986.

Examples

```
data(Vaal.Flow)
```

Index