Package 'LSBP'

June 21, 2020

Type Package

Title Tractable Bayesian density regression via logit stick-breaking priors

Version 0.0.4

Date 2020-06-21

Author Tommaso Rigon

Maintainer Tommaso Rigon <tommaso.rigon@gmail.com>

Description Posterior inference for the logit stick-breaking model with Gaussian kernels. The LSBP package is an implementation of the algorithms described in the paper: Rigon, T. and Durante, D. (2020). Tractable Bayesian Tractable Bayesian density regression via logit stick-breaking priors. Journal of Statistical Planning and Inference.

LazyData TRUE

Imports Rcpp (>= 0.12.9),
 Formula,
 mvtnorm,
 cluster,
 BayesLogit (>= 2.1)

LinkingTo Rcpp, RcppArmadillo

License MIT + file LICENSE

RoxygenNote 6.1.1

R topics documented:

---

control_ECM   *Control parameters for the ECM algorithm*

Description

This auxiliary function can be used for specifying the settings (i.e. maximum number of iterations, tolerance level, initialization method), of the [LSBP_ECM](#) main function.

Usage

```
control_ECM(maxiter = 10000, tol = 0.001, method_init = "cluster")
```

Arguments

maxiter   An integer indicating the maximum number of iterations for the [LSBP_ECM](#) algorithm.

tol   A real number controlling the convergence of the algorithm. The [LSBP_ECM](#) algorithm stops when the difference between consecutive values of the log-posterior is smaller than tol.

method_init   The initialization criterium. By default, method_init='cluster' preallocates covariates into groups using [clara](#). Other available methods are: method_init='random' and method_init='deterministic'.

Value

The function returns a list having the same entries provided as argument. Missing arguments are filled with default values.

---

control_Gibbs   *Control parameters for the Gibbs sampling algorithm*

Description

This auxiliary function can be used for specifying the technical settings (i.e. number of MCMC iterations, burn-in, initialization method), of the [LSBP_Gibbs](#) main function.

Usage

```
control_Gibbs(R = 5000, burn_in = 1000, method_init = "cluster")
```

Arguments

R   An integer indicating the number of replications to be computed after the burn-in.

burn_in   An integer indicating the number of replication discarded as burn-in period.

method_init   The initialization criterium. By default, method_init='cluster' preallocates covariates into groups using [clara](#). Other available possibilities are: method_init='random' and method_init='deterministic'.

Value

The function returns a list having the same entries provided as argument. Missing arguments are filled with default values.

---

control_VB	*Control parameters for the VB algorithm*

Description

This auxiliary function can be used for specifying the technical settings (i.e. maximum number of iterations, tolerance level, initialization method), of the LSBP_VB main function.

Usage

```
control_VB(maxiter = 10000, tol = 0.01, method_init = "cluster")
```

Arguments

maxiter   An integer indicating the maximum number of iterations for the VB algorithm.

tol   A real number controlling the convergence of the algorithm. The LSBP_VB algorithm stops when the difference between consecutive values of the log-posterior is smaller than tol.

method_init   The initialization criterium. By default, method_init='cluster' preallocates covariates into groups using clara. Another available method is method_init='random'.

Value

The function returns a list having the same entries provided as argument. Missing arguments are filled with default values.

LSBP_density	*Conditional density for a LSBP model*

Description

Evaluate the conditional density of a LSBP given the parameters.

Usage

```
LSBP_density(y, X1, X2, beta_kernel, beta_mixing, tau)
```

Arguments

y   A value of which the conditional density has to be computed

X1   A n x p_kernel design matrix for the kernel

X2   A n x p_mixing design matrix for the stick-breaking weights

beta_kernel   A H x p_kernel dimensional matrix of coefficients for the linear predictor of the kernel

beta_mixing   A H-1 x p_mixing dimensional matrix of coefficients for the linear predictor of the stick-breaking weights

tau   A H dimensional vector of coefficients for the kernel precision

Details

The function LSBP_density evaluates the conditional density of y given the parameters

---

LSBP_ECM   *ECM algorithm for the LSBP model*

Description

This function is an implementation of the expectation maximization Algorithm 2 in Rigon, T. and Durante, D. (2020).

Usage

```
LSBP_ECM(Formula, data, H, prior, control = control_ECM(),
  verbose = TRUE)
```

Arguments

Formula	An object of class [Formula]: a symbolic description of the model to be fitted. The details of model specification are given under 'Details'.
data	A data frame containing the variables of Formula. The data frame MUST be provided.
H	An integer indicating the number of mixture components.
prior	A list of prior hyperparameters as returned by [prior_LSBP]. If missing, default prior values are used.
control	A list as returned by [control_ECM].
verbose	A logical value indicating whether additional information should be displayed while the algorithm is running.

Details

The Formula specification contains the response, separated from the covariates with the symbol '~', and two sets of covariates. The latters are separated by the symbol '|', indicating the kernel covariates and the mixing covariates, respectively. For example, one could specify y ~ x1 + x2 | x3 + x4. NOTE: if the second set of covariates is omitted then it is implicitely assumed that the two sets are the same.

If offsets or weights are provided in the Formula they will be ignored in the current version. A predict method is available and described at [predict.LSBP_ECM].

Value

The output is an object of class 'LSBP_ECM' containing the following quantities:

- param. A list containing the maximum a posteriori, for each set of coefficients: beta_mixing,beta_kernel,tau.
- cluster. A n dimensional vector containing, for each observation, the mixture component having with the highest probability.
- z. A n x H matrix containing the probabilities of belonging to each of the mixture components, where n denotes the number of observations.
- logposterior. The log-posterior of the model at convergence.

- call. The input Formula.
- data. The input data frame.
- control. The control list provided as input.
- H. The input number of mixture components.
- prior. The input prior hyperparameters.

References

Rigon, T. and Durante, D., (2020), Tractable Bayesian density regression via logit stick-breaking priors. Journal of Statistical Planning and Inference.

Examples

```
## Not run:
data(cars)

# A model with constant kernels
fit_em <- LSBP_ECM(dist ~  1 | speed, data=cars, H=4)
plot(cars)
lines(cars$speed,predict(fit_em))

# A model with linear kernels
fit_em <- LSBP_ECM(dist ~ speed | speed, data=cars, H=2)
plot(cars)
lines(cars$speed,predict(fit_em))

## End(Not run)
```

LSBP_Gibbs   *Gibbs sampling algorithm for the LSBP model*

Description

This function is an implementation of the Gibbs sampling Algorithm 1 in Rigon, T. and Durante, D. (2020).

Usage

```
LSBP_Gibbs(Formula, data, H, prior, control = control_Gibbs(),
  verbose = TRUE)
```

Arguments

Formula	An object of class [Formula](#): a symbolic description of the model to be fitted. The details of model specification are given under 'Details'.
data	A data frame containing the variables of Formula.
H	An integer indicating the number of mixture components.
prior	A list of prior hyperparameters as returned by [prior_LSBP](#). If missing, default prior values are used.
control	A list as returned by [control_Gibbs](#).
verbose	A logical value indicating whether additional information should be displayed while the algorithm is running.

Details

The Formula specification contains the response, separated from the covariates with the symbol '~', and two sets of covariates. The latters are separated by the symbol '|', indicating the kernel covariates and the mixing covariates, respectively. For example, one could specify y ~ x1 + x2 | x3 + x4. NOTE: if the second set of covariates is omitted it is implicitely assumed that the two sets are the same.

If offsets or weights are provided in the Formula they will be ignored in the current version. A predict method is available and described at predict.LSBP_Gibbs.

Value

The output is an object of class 'LSBP_Gibbs' containing the following quantities:

- param. A list containing MCMC replications for each set of coefficients: beta_mixing,beta_kernel,tau.

- logposterior. The log-posterior of the model at each MCMC iteration.

- call. The input Formula.

- data. The input data frame.

- control. The control list provided as input.

- H. The input number of mixture components.

- prior. The input prior hyperparameters.

References

Rigon, T. and Durante, D., (2020), Tractable Bayesian density regression via logit stick-breaking priors. Journal of Statistical Planning and Inference.

Examples

```
## Not run:
data(cars)

# A model with constant kernels
fit_gibbs <- LSBP_Gibbs(dist ~  1 | speed, data=cars, H=4)
plot(cars)
lines(cars$speed,colMeans(predict(fit_gibbs)))

# A model with linear kernels
fit_gibbs <- LSBP_Gibbs(dist ~ speed | speed, data=cars, H=2)
plot(cars)
lines(cars$speed,colMeans(predict(fit_gibbs)))

## End(Not run)
```

LSBP_VB   *Variational Bayes algorithm for the LSBP model*

Description

This function is an implementation of the variational Bayes Algorithm 3 in Rigon, T. and Durante, D. (2020).

Usage

```
LSBP_VB(Formula, data, H, prior, control = control_VB(),
  verbose = TRUE)
```

Arguments

Formula	An object of class [Formula](#): a symbolic description of the model to be fitted. The details of model specification are given under 'Details'.
data	A data frame containing the variables of Formula.
H	An integer indicating the number of mixture components.
prior	A list of prior hyperparameters as returned by [prior_LSBP](#). If missing, default prior values are used.
control	A list as returned by [control_VB](#).
verbose	A logical value indicating whether additional information should be displayed while the algorithm is running.

Details

The Formula specification contains the response, separated from the covariates with the symbol '~', and two sets of covariates. The latters are separated by the symbol '|', indicating the kernel covariates and the mixing covariates, respectively. For example, one could specify y ~ x1 + x2 | x3 + x4. NOTE: if the second set of covariates is omitted it is implictely assumed that the two sets are the same.

If offsets or weights are provided in the Formula they will be ignored in the current version. A predict method is available and described at [predict.LSBP_VB](#).

Value

The output is an object of class 'LSBP_VB' containing the following quantities:

- param. A list containing the parameters for the variational approximation of each distribution: mu_mixing,Sigma_mixing,mu_kernel,Sigma_kernel,a_tilde,b_tilde.
- cluster. A n dimensional vector containing, for each observation, the mixture component having with the highest probability.
- z. A n x H matrix containing the probabilities of belonging to each of the mixture components, where n denotes the number of observations.
- logposterior. The log-posterior of the model at convergence.
- call. The input Formula
- data. The input data frame.

- control. The control list provided as input.
- H. The input number of mixture components.
- prior. The input prior hyperparameters.

References

Rigon, T. and Durante, D., (2020), Tractable Bayesian density regression via logit stick-breaking priors. Journal of Statistical Planning and Inference.

Examples

```
data(cars)

# A model with constant kernels
fit_vb <- LSBP_VB(dist ~  1 | speed, data=cars, H=4)
plot(cars)
lines(cars$speed,colMeans(predict(fit_vb)))

# A model with linear kernels
fit_vb <- LSBP_VB(dist ~ speed | speed, data=cars, H=2)
plot(cars)
lines(cars$speed,colMeans(predict(fit_vb)))
```

---

predict.LSBP_ECM   *Predict method for the LSBP*

Description

Predict method for a LSBP estimated using the LSBP_ECM function.

Usage

```
## S3 method for class 'LSBP_ECM'
predict(object, type = "mean", newdata = NULL,
  threshold = NULL, ...)
```

Arguments

object	An object of class LSBP_ECM.
type	String indicating the type of prediction: type="mean",type="variance" or type="cdf". See details.
newdata	A new data frame containing the same variables declared in Formula. If missing, the dataset provided for estimation is used.
threshold	Only needed if type="cdf" is selected. See details.
...	Further arguments passed to or from other methods.

Details

The method predict.LSBP_ECM produces predicted values, obtained by evaluating the conditional mean (if type="mean"), the conditional variance (if type="variance") or the conditional cumulative distribution function (if type="cdf") at a given threshold, after plugging-in the MAP, and using the observations contained in the newdata data frame.

---

predict.LSBP_Gibbs *Predict method for the LSBP*

Description

Predict method for a LSBP estimated using the LSBP_Gibbs function.

Usage

```
## S3 method for class 'LSBP_Gibbs'
predict(object, type = "mean", newdata = NULL,
  threshold = NULL, ...)
```

Arguments

object	An object of class LSBP_Gibbs.
type	String indicating the type of prediction: type="mean",type="predictive",type="variance" or type="cdf". See details.
newdata	A new data frame containing the same variables declared in Formula. If missing, the dataset provided for estimation is used.
threshold	Only needed if type="cdf" is selected. See details.
...	Further arguments passed to or from other methods.

Details

The method predict.LSBP_Gibbs produces a sample of predicted values, obtained by evaluating the conditional mean of the LSBP model or the predictive distribution, using the observations contained in the newdata data frame.

If type="mean" a sample from the posterior distribution of the LSBP mean is returned. If type="predictive" is selected, then a sample from the predictive distribution is returned. If type="variance" a sample from the posterior distribution of the LSBP variance is returned. If type="cdf" a sample from the posterior distribution of the LSBP cumulative distribution function is returned, evaluated at threshold.

---

predict.LSBP_VB *Predict method for the LSBP*

Description

Predict method for a LSBP estimated using the LSBP_VB function.

Usage

```
## S3 method for class 'LSBP_VB'
predict(object, type = "mean", R = 5000,
  newdata = NULL, threshold = NULL, ...)
```

Arguments

`object`	An object of class [`LSBP_VB`].
`type`	String indicating the type of prediction: `type="mean"`,`type="predictive"`, `type="variance"` or `type="cdf"`. See details.
`R`	An integer indicating the number of replications for the returned sample.
`newdata`	A new data frame containing the same variables declared in `Formula`. If missing, the dataset provided for estimation is used.
`threshold`	Only needed if `type="cdf"` is selected. See details.
`...`	Further arguments passed to or from other methods.

Details

The method `predict.LSBP_VB` produces a sample of predicted values, obtained by evaluating the conditional mean of the LSBP model or the predictive distribution, using the observations contained in the `newdata` data frame.

If `type="mean"` a sample from the posterior distribution of the LSBP mean is returned. If `type="predictive"` is selected, then a sample from the predictive distribution is returned. If `type="variance"` a sample from the posterior distribution of the LSBP variance is returned. If `type="cdf"` a sample from the posterior distribution of the LSBP cumulative distribution function is returned, evaluated at `threshold`.

---

prior_LSBP	*Prior specification for the LSBP model*

Description

This auxiliary function can be used for specifying the prior hyperparameters in the [`LSBP_Gibbs`], [`LSBP_ECM`], [`LSBP_VB`] main functions.

Usage

```
prior_LSBP(p_kernel, p_mixing, b_kernel = rep(0, p_kernel),
  B_kernel = diag(10^6, p_kernel), b_mixing = rep(0, p_mixing),
  B_mixing = diag(10^4, p_mixing), a_tau = 0.1, b_tau = 0.1)
```

Arguments

`p_kernel, p_mixing`	
	The dimension of the design matrices for the kernel component and the mixing component, respectively.
`b_kernel`	A `p_kernel` dimensional vector representing the prior mean for the Gaussian kernel coefficients.
`B_kernel`	A `p_kernel` x `p_kernel` matrix representing the prior covariance of the Gaussian kernel coefficients.
`b_mixing`	A `p_mixing` dimensional vector containing the prior mean of the Gaussian mixing coefficients
`B_mixing`	A `p_mixing` x `p_mixing` matrix representing the prior covariance of the Gaussian mixing coefficients.
`a_tau, b_tau`	The hyperparameters of a Gamma prior distribution for the kernel precision.

Value

The function returns a list having the same entries provided as argument. Missing arguments are filled with default values, although this is NOT recommended in general.

Examples

```
## Not run:
data(cars)
prior  <- prior_LSBP(p_kernel=1, p_mixing=2, a_tau=1.5 ,b_tau=1.5)
fit_em <- LSBP_ECM(dist ~ 1 | speed, data=cars, H=4, prior=prior)

## End(Not run)
```

Index