Package 'LSBP'

June 22, 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 Bayesian density regression via logit stick-breaking priors. The LSBP package is an implementation of the algorithms described in: Rigon, T. and Durante, D. (2020).

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

The control_ECM function can be used for specifying the technical settings (i.e. the maximum number of iterations, the tolerance level, and the initialization method), of the LSBP_ECM 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 method. The default method_init='cluster' partitions the covariates using the clara clustering algorithm. Other available options are: method_init='random' and method_init='deterministic'.

Value

The inputs are converted into a list. Missing arguments are filled with default values.

control_Gibbs *Control parameters for the Gibbs sampling algorithm*

Description

The control_Gibbs function can be used for specifying the technical settings (i.e. the number of MCMC iterations, the burn-in, and the initialization method), of the LSBP_Gibbs function.

Usage

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

Arguments

R	An integer indicating the number of MCMC iterations to be computed after the burn-in.
burn_in	An integer indicating the number of MCMC iterations discarded as burn-in period.
method_init	The initialization method. The default method_init='cluster' partitions the covariates using the clara clustering algorithm. Other available options are: method_init='random' and method_init='deterministic'.

Value

The inputs are converted into a `list`. Missing arguments are filled with default values.

`control_VB` *Control parameters for the VB algorithm*

Description

The `control_VB` function can be used for specifying the technical settings (i.e. the maximum number of iterations, the tolerance level, and the 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 evidence lower bound (ELBO) is smaller than `tol`.
`method_init`	The initialization method. The default `method_init='cluster'` partitions the covariates using the `clara` clustering algorithm. Another available option is `method_init='random'`.

Value

The inputs are converted into a `list`. Missing arguments are filled with default values.

`LSBP_density` *Conditional density of a LSBP model*

Description

Evaluate the conditional density $f_x(y)$ of a LSBP model, given the parameters and the covariates.

Usage

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

Arguments

`y`	The value at which the conditional density must be evaluated.
`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 $f_x(y)$. The number of mixture components H is inferred from the dimensions of beta_mixing and beta_kernel.

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 estimated. The details of model specification are given under "Details".
data	A data frame containing the variables described in 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, although this is NOT recommended.
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 y, 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, and 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. NOTE: the log-posterior is reported up to an additive constant.
- 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 estimated. The details of model specification are given under "Details".
data	A data frame containing the variables described in 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, although this is NOT recommended.
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 y, 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 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. NOTE: the log-posterior is reported up to an additive constant.
- 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))) # Posterior mean
```

```
# 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))) # Posterior mean

## 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 estimated. The details of model specification are given under "Details".
data	A data frame containing the variables described in 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, although this is NOT recommended.
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 y, 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 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.
- `lowerbound`. The `lowerbound` is the evidence lower bound (ELBO) of the model at convergence. NOTE: the `lowerbound` is reported up to an additive constant.
- `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 model, 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. The available options are: 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 maximum a posteriori, and using the observations contained in the newdata data frame.

---

predict.LSBP_Gibbs *Predict method for the LSBP*

Description

Predict method for a LSBP model 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. The available options are 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", then a sample from the posterior of the mean of a LSBP model is returned. If type="predictive" is selected, then a sample from the predictive distribution is returned. If type="variance", then a sample from the posterior distribution of the LSBP variance is returned. If type="cdf", then 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 model 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", then a sample from the (variational) posterior of the mean of a LSBP model is returned. If type="predictive" is selected, then a sample from the (variational) predictive distribution is returned. If type="variance", then a sample from the (variational) posterior distribution of the LSBP variance is returned. If type="cdf", then a sample from the (variational) 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