Package ‘probe’

August 11, 2022

Type Package

Title PaRtitiOned empirical Bayes Ecm (PROBE) algorithm

Version 1.0

Date 2022-08-31

Author Alex McLain

Maintainer Anja Zgodic <azgodic@email.sc.edu>

Description This package contains functions to fit an efficient and powerful Bayesian approach for sparse high-dimensional linear regression. Unlike previous approaches, our model formulation is focused on maximum a posteriori probability (MAP) estimation of the prior mean of the regression coefficients which have a traditional spike-and-slab form. Minimal prior assumptions on the parameters are required through the use of plug-in empirical Bayes estimates of hyperparameters. Efficient estimation is completed through the use of a partitioned and extended expectation conditional maximization (ECM) algorithm. The result is a PaRtitiOned empirical Bayes Ecm (PROBE) algorithm applied to sparse high-dimensional linear regression. We give functions to estimate credible and prediction intervals for predictions of future values.

License GPL (>= 2)

Encoding UTF-8

RoxygenNote 7.1.2

Imports Rcpp (>= 1.0.6)

LinkingTo Rcpp, RcppArmadillo

NeedsCompilation yes

Depends R (>= 4.00)

R topics documented:

probe-package *PaRtitiOned empirical Bayes Ecm (PROBE) algorithm*

Description

This package contains functions to fit an efficient and powerful Bayesian approach for sparse high-dimensional linear regression. Unlike previous approaches, our model formulation is focused on maximum a posteriori probability (MAP) estimation of the prior mean of the regression coefficients which have a traditional spike-and-slab form. Minimal prior assumptions on the parameters are required through the use of plug-in empirical Bayes estimates of hyperparameters. Efficient estimation is completed through the use of a partitioned and extended expectation conditional maximization (ECM) algorithm. The result is a PaRtitiOned empirical Bayes Ecm (PROBE) algorithm applied to sparse high-dimensional linear regression. We give functions to estimate credible and prediction intervals for predictions of future values.

Package Content

Index of help topics:

```
Sim_data                Simulated high-dimensional data set for sparse
                        linear regression.
Sim_data_cov            Simulated high-dimensional data set for sparse
                        linear regression with non-sparse covariates.
e_step_func             Function for fitting the empirical Bayes
                        portion of the E-step
m_step_regression       Function for fitting the initial part of the
                        M-step
predict_probe_func      Obtaining predictions, confidence intervals and
                        prediction intervals from probe
probe                   Fitting PaRtitiOned empirical Bayes Ecm (PROBE)
                        algorithm to sparse high-dimensional linear
                        models.
probe-package           PaRtitiOned empirical Bayes Ecm (PROBE)
                        algorithm
```

Maintainer

Anja Zgodic <azgodic@email.sc.edu>

Author(s)

Alex McLain

e_step_func — *Function for fitting the empirical Bayes portion of the E-step*

Description

A wrapper function estimating posterior expectations of the γ variables using an empirical Bayesian technqiue.

Usage

```
e_step_func(beta_t, beta_var, df, adj = 3, lambda = 0.1, monotone = TRUE)
```

Arguments

`beta_t`	Expectation of the posterior mean (assuming $\gamma = 1$)
`beta_var`	Current posterior variance (assuming $\gamma = 1$)
`df`	Degrees of freedom for the t-distribution (use to calculate p-values).
`adj`	bandwidth multiplier to Silverman's 'rule of thumb' for calculating the marginal density of the test-statistics (default = 3).
`lambda`	value of the λ parameter for estimating the proportion of null hypothesis using Storey et al. (2004) (default = 0.1).
`monotone`	Logical - Should the estimated marginal density of the test-statistics be monotone non-increasing from zero (default = TRUE).

Value

A list including

`delta` estimated posterior expectations of the γ.

`pi0` estimated proportion of null hypothesis

Examples

```
#not run
#mod <- e_step_func(beta_t, beta_var, df, adj = 3, lambda = 0.1, monotone = TRUE)
```

m_step_regression — *Function for fitting the initial part of the M-step*

Description

A wrapper function providing the quantities related to the M-step for α_0 and σ^2.

Usage

```
m_step_regression(Y, W, W2, X, a = -3/2, Int = TRUE)
```

Arguments

`Y`	A matrix containing the outcome `Y`
`W`	Quantity $E(W_0)$ as outlined in citation, output from W_update_fun
`W2`	Quantity $E(W_0^2)$ as outlined in citation, output from W_update_fun
`X`	A matrix or dataframe of other predictors to account for
`a`	(optional) parameter for changing the hyperparameter a (default, $a = -3/2$ uses $n-2$ as denominator for MAP of σ^2)
`Int`	(optional) Logical - should an intercept be used?

Value

A list including

`coef` the MAP estimates of the α_0 parameters `sigma2_est` the MAP estimate of σ^2 `VCV` posterior variance covariance matrix of α_0, `res_data` dataframe containing MAP estimates, posterior variances, t-test statistics and associated p-values for α_0

Examples

```
#not run
#mod <- m_step_regression(Y, W_ast, W_ast_var + W_ast^2, X)
```

---

predict_probe_func — *Obtaining predictions, confidence intervals and prediction intervals from probe*

Description

A function providing predictions, along with $(1-\alpha) * 100\%$ credible, and prediction intervals for new observations.

Usage

```
predict_probe_func(res, Z, X, alpha, Z_2)
```

Arguments

`res`	The results from the probe function.
`Z`	A matrix containing the predictors on which to apply the probe algorithm
`X`	(optional) A matrix or dataframe of predictors not subjected to the sparsity assumption to account for.
`alpha`	Type I error; significance level
`Z_2`	(optional) Square of Z matrix.

Value

A dataframe with predictions, confidence intervals, and prediction intervals for each new observation.

Examples

```
### Example
data(Sim_data)
attach(Sim_data)
alpha <- 0.05
plot_ind <- TRUE

# Run the analysis. Y_test and Z_test are included for plotting purposes only
full_res <- probe( Y = Y, Z = Z, alpha = alpha, plot_ind = plot_ind, Y_test = Y_test, Z_test = Z_test)

# Predicting for test data
pred_res <- predict_probe_func(full_res, Z = Z_test, X = NULL, alpha = alpha)
head(pred_res)
```

---

probe *Fitting PaRtitiOned empirical Bayes Ecm (PROBE) algorithm to sparse high-dimensional linear models.*

Description

A wrapper function for the main PROBE algorithm function. The R package is a work in progress.

Usage

```
probe(Y, Z, X, alpha, ep, maxit, Y_test, Z_test, X_test, verbose,
signal, eta_i, plot_ind)
```

Arguments

`Y`	The outcome variable.
`Z`	An `n x M` matrix of sparse predictors variables.
`X`	(optional) An `n x p` matrix or dataframe of other predictors not subjected to the sparsity assumption.
`alpha`	Type I error; significance level
`ep`	Value against which to compare convergence criterion (default = 0.1).
`maxit`	Maximum number of iterations the algorithm will run for (default = 10000).
`Y_test`	(optional) Test Y data used plotting purposes only (doesn't impact results).
`Z_test`	(optional) Test Z data used plotting purposes only (doesn't impact results).
`X_test`	(optional) Test X data used plotting purposes only (doesn't impact results).
`verbose`	A logical (true/false) value whether to print algorithm iteration progress and summary quantities (default = FALSE).
`signal`	(optional) A vector of indicies of the true non-null coefficients. This is used to calculate the true and false discovery rates by iteration for simulated data. Used plotting purposes only (doesn't impact results).
`eta_i`	(optional) A vector of the true signal. This is used to calculate the MSE by iteration for simulated data. Used plotting purposes only (doesn't impact results).
`plot_ind`	A logical values (True/False) for whether to output plots on algorithm results and progress (default = FALSE)

Value

A list including

`beta_ast_hat` MAP estimates of the regression coefficients (β^*),

`beta_hat,beta_hat_var` MAP estimates of the posterior expectation (beta_hat) and variance (beta_hat_var) of the prior mean (β) of the regression coefficients assuming $\gamma = 1$,

`gamma_hat` the posterior expectation of the latent γ variables,

`sigma2_est` MAP estimate of the residual variance,

`E_step` full results of the final E_step,

`Calb_mod` results of first (α_0) part of the M-step,

`count` the total number of iterations before convergence.

See Also

predict_probe_func to obtain predictions, credible intervals and prediction intervals from PROBE.

Examples

```
### Example
data(Sim_data)
attach(Sim_data)
alpha <- 0.05
plot_ind <- TRUE

# Run the analysis. Y_test and Z_test are included for plotting purposes only
full_res <- probe( Y = Y, Z = Z, alpha = alpha, Y_test = Y_test,
Z_test = Z_test, plot_ind = plot_ind)

# Predicting for test data
pred_res <- predict_probe_func(full_res, Z = Z_test, alpha = alpha)
head(pred_res)

# Estimate of the residual variance
full_res$sigma2_est

### Example with additional covariate data X (not subjected to the sparsity assumption)
data(Sim_data_cov)
attach(Sim_data_cov)

# Calculating the true signal (the impact of Z only)
eta_i <- apply(t(Z)*beta_tr,2,sum)
# Run the analysis. eta_i (true signal) and signal are included for plotting purposes only.
full_res <- probe( Y = Y, Z = Z, X = X, alpha = alpha,
signal = signal, eta_i  = eta_i, plot_ind = plot_ind)

# Final estimates of the impact of X versus the true values:
data.frame(true_values = beta_X_tr, full_res$Calb_mod$res_data[-2,])

#Compare to a standard linear model of X on Y:
summary(lm(Y~X$Cont_cov + X$Binary_cov))$coefficients
```

Sim_data — *Simulated high-dimensional data set for sparse linear regression.*

Description

This dataset was simulated using a 100×100 2-dimensional setting described in the reference. The data contains 400 subjects with one outcome and 10,000 predictor variables. There is also test outcomes and predictor variables.

Usage

```
data("Sim_data")
```

Format

A data frame with 400 observations and the following objects:

`Y` Outcome variable of length 400.

`Z` A 400×10000 matrix of binary predictor variables.

`beta_tr` The true values of all 10000 regression coefficients.

`signal` The locations of the non-zero regression coefficients.

`Y_test` Outcome variable of length 400 for test set.

`Z_test` A 400×10000 matrix of binary predictor variables for test set.

References

To come

Examples

```
data(Sim_data)
attach(Sim_data)
length(Y)
dim(Z)
```

Sim_data_cov — *Simulated high-dimensional data set for sparse linear regression with non-sparse covariates.*

Description

This dataset was simulated using a 100×100 2-dimensional setting described in the reference only two covariates are added. The data contains 400 subjects with one outcome, 10,000 predictor variables which are to be subjected to the sparsity assumption, and 2 covariates which are not to be subjected to the sparsity assumption.

Usage

```
data("Sim_data_cov")
```

Format

A data frame with 400 observations and the following objects:

- `Y` Outcome variable of length 400.
- `X` A dataframe of a continuous (`Cont_cov`) and binary (`Binary_cov`) covariate.
- `Z` A 400×10000 matrix of binary predictor variables.
- `beta_tr` The true values of all 10000 regression coefficients.
- `beta_X_tr` The true values of the intercept, `Cont_cov`, and `Binary_cov`.
- `signal` The locations of the non-zero regression coefficients.

References

To come

Examples

```
data(Sim_data_cov)
attach(Sim_data_cov)
length(Y)
summary(X)
dim(Z)
```

Index