Package ‘probe’

October 31, 2023

Type Package

Title Sparse High-Dimensional Linear Regression with PROBE

Version 1.1

Date 2023-10-01

Description Implements an efficient and powerful Bayesian approach for sparse high-dimensional linear regression. It uses minimal prior assumptions on the parameters through plug-in empirical Bayes estimates of hyperparameters. An efficient Parameter-Expanded Expectation-Conditional-Maximization (PX-ECM) algorithm estimates maximum a posteriori (MAP) values of regression parameters and variable selection probabilities. The PX-ECM results in a robust computationally efficient coordinate-wise optimization, which adjusts for the impact of other predictor variables. The E-step is motivated by the popular two-group approach to multiple testing. The result is a PaRtitiOned empirical Bayes Ecm (PROBE) algorithm applied to sparse high-dimensional linear regression, implemented using one-at-a-time or all-at-once type optimization. More information can be found in McLain, Zgodic, and Bondell (2022) <arXiv:2209.08139>.

BugReports https://github.com/alexmclain/PROBE/issues

License GPL (>= 2)

Encoding UTF-8

RoxygenNote 7.2.3

Imports Rcpp, glmnet

LinkingTo Rcpp, RcppArmadillo

NeedsCompilation yes

Depends R (>= 4.00)

Author Alexander McLain [aut, cre] (<https://orcid.org/0000-0002-5475-0670>), Anja Zodiac [aut, ctb]

Maintainer Alexander McLain <mclaina@mailbox.sc.edu>

Repository CRAN

Date/Publication 2023-10-31 16:30:02 UTC

R topics documented:

`probe-package` *probe: Sparse High-Dimensional Linear Regression with PROBE*

Description

Implements an efficient and powerful Bayesian approach for sparse high-dimensional linear regression. It uses minimal prior assumptions on the parameters through plug-in empirical Bayes estimates of hyperparameters. An efficient Parameter-Expanded Expectation-Conditional-Maximization (PX-ECM) algorithm estimates maximum a posteriori (MAP) values of regression parameters and variable selection probabilities. The PX-ECM results in a robust computationally efficient coordinate-wise optimization, which adjusts for the impact of other predictor variables. The E-step is motivated by the popular two-group approach to multiple testing. The result is a PaRtitiOned empirical Bayes Ecm (PROBE) algorithm applied to sparse high-dimensional linear regression, implemented using one-at-a-time or all-at-once type optimization. More information can be found in McLain, Zgodic, and Bondell (2022) arXiv:2209.08139.

Details

Examples for applying PROBE to sparse high-dimensional linear regression are given for one-at-a-time `probe_one` or all-at-once `probe` type optimization.

Author(s)

Maintainer: Alexander McLain `<mclaina@mailbox.sc.edu>` (ORCID)

Authors:

- Anja Zodiac [contributor]

References

- McLain, A. C., Zgodic, A., & Bondell, H. (2022). Sparse high-dimensional linear regression with a partitioned empirical Bayes ECM algorithm. arXiv preprint arXiv:2209.08139.

See Also

Useful links:

- Report bugs at https://github.com/alexmclain/PROBE/issues

`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 = 5, 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 = 5).
`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

References

Storey, J. D., Taylor, J. E., and Siegmund, D. (2004), "Strong control, conservative point estimation and simultaneous conservative consistency of false discovery rates: A unified approach," J. R. Stat. Soc. Ser. B. Stat. Methodol., 66, 187–205. McLain, A. C., Zgodic, A., & Bondell, H. (2022). Sparse high-dimensional linear regression with a partitioned empirical Bayes ECM algorithm. arXiv preprint arXiv:2209.08139.

`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, Z = NULL, 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
`Z`	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

References

McLain, A. C., Zgodic, A., & Bondell, H. (2022). Sparse high-dimensional linear regression with a partitioned empirical Bayes ECM algorithm. arXiv preprint arXiv:2209.08139.

`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, X, Z = NULL, alpha = 0.05, X_2 = NULL)
```

Arguments

- `res` The results from the probe function.
- `X` A matrix containing the predictors on which to apply the probe algorithm
- `Z` (optional) A matrix or dataframe of predictors not subjected to the sparsity assumption to account for.
- `alpha` significance level for $(100(1-\alpha)\%)$ credible and prediction intervals.
- `X_2` (optional) Square of X matrix.

Value

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

References

McLain, A. C., Zgodic, A., & Bondell, H. (2022). Sparse high-dimensional linear regression with a partitioned empirical Bayes ECM algorithm. arXiv preprint arXiv:2209.08139. Zgodic, A., Bai, R., Zhang, J., Wang, Y., Rorden, C., & McLain, A. (2023). Heteroscedastic sparse high-dimensional linear regression with a partitioned empirical Bayes ECM algorithm. arXiv preprint arXiv:2309.08783.

Examples

```
### Example
data(Sim_data)
data(Sim_data_test)
attach(Sim_data)
attach(Sim_data_test)
alpha <- 0.05
plot_ind <- TRUE
adj <- 10

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

# Predicting for test data
pred_res <- predict_probe_func(full_res, X = X_test, alpha = alpha)
sqrt(mean((Y_test - pred_res$Pred)^2))
head(pred_res)
```

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

Description

A wrapper function for the all-at-once variant of the PROBE algorithm.

Usage

```
probe(Y, X, Z = NULL, ep = 0.1, maxit = 10000, Y_test = NULL, X_test = NULL,
Z_test = NULL, verbose = FALSE, signal = NULL, eta_i = NULL, alpha = 0.05,
plot_ind = FALSE, adj = 5)
```

Arguments

Y	The outcome variable.
X	An n x M matrix of sparse predictors variables.
Z	(optional) An n x p matrix or dataframe of other predictors not subjected to the sparsity assumption.
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).
X_test	(optional) Test X data used plotting purposes only (doesn't impact results).
Z_test	(optional) Test Z 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).
alpha	(optional) significance level
plot_ind	A logical values (True/False) for whether to output plots on algorithm results and progress (default = FALSE)
adj	Bandwidth parameter for empirical Bayes E-step. The bandwidth will be equal to adj times Silverman's 'rule of thumb' (default = 2).

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.

References

McLain, A. C., Zgodic, A., & Bondell, H. (2022). Sparse high-dimensional linear regression with a partitioned empirical Bayes ECM algorithm. arXiv preprint arXiv:2209.08139..

See Also

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

Examples

```
### Example
data(Sim_data)
data(Sim_data_test)
attach(Sim_data)
attach(Sim_data_test)
alpha <- 0.05
plot_ind <- TRUE
adj <- 10

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

# Predicting for test data
pred_res <- predict_probe_func(full_res, X = X_test)
sqrt(mean((Y_test - pred_res$Pred)^2))

# Estimate of the residual variance and true value
full_res$sigma2_est
sigma2_tr

# RMSE of estimated beta coefficients
beta_ast_est <- full_res$beta_ast_hat
sqrt(mean((beta_ast_est - beta_tr)^2))

# Posterior expectation of gamma by true
gamma_est <- full_res$E_step$gamma
sum(gamma_est)
sum(gamma_est[beta_tr>0])

### Examples that incorporate additional covariate data (Z)
### not subjected to the sparsity assumption available on Github.
```

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

Description

A wrapper function for the one-at-a-time variant of the PROBE algorithm.

Usage

```
probe_one(Y, X, ep = 0.001, maxit = 10000, Y_test = NULL, X_test = NULL,
verbose = FALSE, signal = NULL, eta_i = NULL, alpha = 0.05, plot_ind = FALSE,
order.method = "lasso", adj = 10, delta = 0.4, update_order= NULL, beta_start= NULL,
seed = NULL)
```

Arguments

`Y`	The outcome variable.
`X`	An `n x M` matrix of sparse predictors variables.
`ep`	Value against which to compare convergence criterion (default = 0.001).
`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).
`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).
`alpha`	(optional) significance level
`plot_ind`	A logical values (True/False) for whether to output plots on algorithm results and progress (default = FALSE)
`order.method`	Updating order and initial values of the algorithm. For `lasso` (default) or `ridge`, a lasso or a ridge regression model (fit with 10-fold CV) will be fitted and used. The `update_order` is defined by the absolute values of the coefficient and `beta_start` is the coefficient values. When using `none`, `update_order` and `beta_start` must be given. `random` will randomly select the updating order and use very small values for `beta_start`.
`adj`	Bandwidth parameter for empirical Bayes E-step. The bandwidth will be equal to `adj` times Silverman's 'rule of thumb' (default = 10).
`delta`	Learning rate for iteration t is (1 + t)^(-1 + delta) (default delta = 0.4).
`update_order`	Manual value for the updating order for when `order.method = "none"` is used.
`beta_start`	Manual value for the starting beta coefficients for when `order.method = "none"` is used.
`seed`	Seed value to ensure reproducibility when `order.method = "lasso"`, `order.method = "ridge"`, or `order.method = "random"`.

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,

`count` the total number of iterations before convergence.

References

McLain, A. C., Zgodic, A., & Bondell, H. (2022). Sparse high-dimensional linear regression with a partitioned empirical Bayes ECM algorithm. arXiv preprint arXiv:2209.08139..

See Also

predict_probe_func to obtain predictions.

Examples

```
### Example
data(Sim_data)
data(Sim_data_test)
attach(Sim_data)
attach(Sim_data_test)
plot_ind <- TRUE
adj <- 10

# Run the analysis. Y_test and X_test are included for plotting purposes only
full_res <- probe_one( Y = Y, X = X, Y_test = Y_test, order.method = "random",
X_test = X_test, plot_ind = plot_ind, adj = adj)

# Predicting for test data
pred_res <- predict_probe_func(full_res, X = X_test)
sqrt(mean((Y_test - pred_res$Pred)^2))

# Estimate of the residual variance and true value
full_res$sigma2_est
sigma2_tr

# RMSE of estimated beta coefficients
beta_ast_est <- c(full_res$beta_ast_hat)
sqrt(mean((beta_ast_est - beta_tr)^2))

# Posterior expectation of gamma by true
gamma_est <- full_res$E_step$gamma
table(gamma_est > 0.5, beta_tr > 0)
sum(gamma_est)
```

```
sum(gamma_est[beta_tr>0])
```

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

Description

This dataset was simulated using a 20×20 2-dimensional setting described in the reference. The data contains 400 subjects with one outcome and 400 predictor variables. The test outcomes and predictor variables are contained in `Sim_data_test`.

Usage

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

Format

A data frame with 400 observations and the following objects:

`Y` Outcome variable of length 400.

`X` A 400×400 matrix of binary predictor variables.

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

`beta_tr` The true values of all 400 regression coefficients.

`sigma2_tr` The true value of the residual variance.

Source

Simulated data.

Examples

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

`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, 10000 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.

`Z` A dataframe of a continuous (`Cont_cov`) and binary (`Binary_cov`) covariate.

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

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

`beta_Z_tr` The true values of the intercept, `Cont_cov`, and `Binary_cov`.

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

Examples

```
data(Sim_data_cov)
str(Sim_data_cov)
```

`Sim_data_test` *Simulated high-dimensional test data set for sparse linear regression*

Description

A test set of outcomes and predictor variables to be used with `Sim_data`.

Usage

```
data("Sim_data_test")
```

Format

A data frame with 400 observations and the following objects:

- `Y_test` Outcome variable of length 400 for test set.
- `Z_test` A 400×400 matrix of binary predictor variables for test set.

Source

Simulated data.

Examples

```
data(Sim_data_test)
attach(Sim_data_test)
length(Y_test)
dim(X_test)
```

Index