Package ‘sparsevb’

October 5, 2020

Type Package

Title Spike-and-Slab Variational Bayes for Linear and Logistic Regression

Version 0.1.0

Date 2020-09-29

Author Gabriel J. Clara [aut, cre], Botond Szabo [aut], Kolyan Ray [aut]

Maintainer Gabriel J. Clara <gabriel.j.clara@gmail.com>

Description Implements variational Bayesian algorithms to perform scalable variable selection for sparse, high-dimensional linear and logistic regression. Features include a novel prioritized updating scheme, which computes a preliminary estimator for the variational means during initialization and generates an update order prioritizing large coefficients. Sparsity is induced via spike-and-slab priors with either Laplace or Gaussian slabs. By default, the heavier-tailed Laplace density is used.

BugReports https://gitlab.com/gclara/varpack/-/issues

Imports Rcpp (>= 1.0.5), selectiveInference (>= 1.2.5)

LinkingTo Rcpp, RcppArmadillo, RcppEnsmallen

SystemRequirements C++11

Encoding UTF-8

RoxygenNote 7.1.1

R topics documented:

`sparsevb-package` *sparsevb: Spike-and-Slab Variational Bayes for Linear and Logistic Regression*

Description

Implements variational Bayesian algorithms to perform scalable variable selection for sparse, high-dimensional linear and logistic regression. Features include a novel prioritized updating scheme, which computes a preliminary estimator for the variational means during initialization and generates an update order prioritizing large coefficients. Sparsity is induced via spike-and-slab priors with either Laplace or Gaussian slabs. By default, the heavier-tailed Laplace density is used.

Details

For details as they pertain to using the package, consult the `svb.fit` function help page. Detailed descriptions and derivations of the variational algorithms with Laplace slabs may be found in the references.

Author(s)

Maintainer: Gabriel J. Clara `<gabriel.j.clara@gmail.com>`

Authors:

- Botond Szabo
- Kolyan Ray

References

- Ray K. and Szabo B. Variational Bayes for high-dimensional linear regression with sparse priors. (2019). *arXiv: 1904.07150 [stat.ME]*.
- Ray K., Szabo B., and Clara G. J. Spike and slab variational Bayes for high dimensional logistic regression. (2020). *Advances in Neural Information Processing Systems 33*.

See Also

Useful links:

- Report bugs at `https://gitlab.com/gclara/varpack/-/issues`

`svb.fit` *Fit Approximate Posteriors to Sparse Linear and Logistic Models*

Description

Main function of the `sparsevb` package. Computes mean-field posterior approximations for both linear and logistic regression models, including variable selection via sparsity-inducing spike and slab priors.

Usage

```
svb.fit(
  X,
  Y,
  family = c("linear", "logistic"),
  slab = c("laplace", "gaussian"),
  mu,
  sigma,
  gamma,
  alpha = 1,
  beta,
  lambda = 1,
  update_order,
  prioritized_init = TRUE,
  exact_math = FALSE,
  rescale = TRUE,
  ridge_penalty = 0.1,
  max_iter = 1000,
  tol = 1e-05
)
```

Arguments

`X` A numeric design matrix, each row of which represents a data point.

`Y` An `nrow(X)`-dimensional response vector, numeric if `family = "linear"` and binary if `family = "logistic"`.

`family` A character string selecting the regression model, either `"linear"` or `"logistic"`. (*default:* `"linear"`)

`slab` A character string specifying the prior slab density, either `"laplace"` or `"gaussian"`. (*default:* `"laplace"`)

`mu` An `ncol(X)`-dimensional numeric vector, serving as initial guess for the variational means. (*default:* `rep(0,ncol(X))`)

`sigma` A positive `ncol(X)`-dimensional numeric vector, serving as initial guess for the variational standard deviations. (*default:* `rep(1,ncol(X))`)

`gamma` An `ncol(X)`-dimensional vector of probabilities, serving as initial guess for the variational inclusion probabilities. (*default:* `rep(alpha/(alpha+beta),ncol(X))`)

`alpha` A positive numeric value, used by the Beta-hyperprior. (*default:* `1.0`)

`beta` A positive numeric value, used by the Beta-hyperprior. (*default:* `ncol(X)`)

`lambda` A numeric value, controlling the scale parameter of the prior slab density. Used as the inverted scale parameter when `prior = "laplace"` (default) and the standard deviation if `prior = "gaussian"`. (*default:* `1.0`)

`update_order` A permutation of `1:ncol(X)`, giving the update order of the coordinate-ascent algorithm. Setting this parameter when `prioritized_init = TRUE` has no effect; it will be overwritten. (*default:* `1:ncol(X)`)

`prioritized_init` A Boolean value, controlling whether the ridge regression estimator for `mu` should be computed during initialization. When `TRUE`, the argument `mu` serves as initial guess for the ridge regression estimator and `update_order` is overwritten by ranking the elements the estimator according to magnitude. (*default:* `TRUE`)

- `exact_math` A Boolean variable, controlling if the linear ridge regression estimator should be computed in closed form or iteratively. Has no effect when `family = "logistic"`. (*default:* `FALSE`)
- `rescale` A Boolean variable, controlling if `X` and `Y` should be rescaled by the estimated variance of the underlying noise. Has no effect when `family = "logistic"`. (*default:* `TRUE`)
- `ridge_penalty` A positive numerical value, controlling the importance of the penalty term when computing the ridge regression estimator. (*default:* `0.1`)
- `max_iter` A positive integer, controlling the maximum number of iterations for the variational update loop. (*default:* `1000`)
- `tol` A positive numerical value, controlling the termination criterion for maximum absolute differences between binary entropies of successive iterates. (*default:* `10e-6`)

Details

Suppose θ is the p-dimensional true parameter. The spike-and-slab prior for θ may be represented by the hierarchical scheme

$$w \sim \mathrm{Beta}(\alpha, \beta)$$

$$z_j \mid w \sim_{i.i.d.} \mathrm{Bernoulli}(w)$$

$$\theta_j \mid z_j \sim_{ind.} (1 - z_j)\delta_0 + z_j g.$$

As usual, δ_0 represents the Dirac measure at 0. The slab g may be taken either as a $\mathrm{Laplace}(0, \lambda^{-1})$ or $N(0, \lambda^2)$ density.

Value

The approximate mean-field posterior, given as a named list containing numeric vectors `"mu"`, `"sigma"`, and `"gamma"`. In mathematical terms,

$$\theta_j \mid \mu_j, \sigma_j, \gamma_j \sim_{ind.} \gamma_j N(\mu_j, \sigma^2) + (1 - \gamma_j)\delta_0.$$

Examples

```
## Not run:

### Simulate a linear regression problem of size n times p, with sparsity level s ###

n <- 2500
p <- 5000
s <- 25

### Generate toy data ###

X <- matrix(rnorm(n*p), n, p) #standard Gaussian design matrix

theta <- numeric(p)
theta[sample.int(p, s)] <- runif(s, -3, 3) #sample non-zero coefficients in random locations

pos_TR <- as.numeric(theta != 0) #true positives

Y <- X %*% theta + rnorm(n) #add standard Gaussian noise
```

```
### Run the algorithm in linear mode with Laplace prior and prioritized initialization ###

test <- svb.fit(X, Y, family = "linear")

posterior_mean <- test$mu * test$gamma #approximate posterior mean

pos <- as.numeric(test$gamma > 0.5) #significant coefficients

### Assess the quality of the posterior estimates ###

TPR <- sum(pos[which(pos_TR == 1)])/sum(pos_TR) #True positive rate

FDR <- sum(pos[which(pos_TR != 1)])/max(sum(pos), 1) #False discovery rate

L2 <- sqrt(sum((posterior_mean - theta)^2)) #L_2-error

MSPE <- sqrt(sum((X %*% posterior_mean - Y)^2)/n) #Mean squared prediction error

## End(Not run)
```

Index