Package 'BayesCOOP'

October 26, 2025

Type Package

Title Implements the BayesCOOP Methodology for Supervised Multimodal Integration

Version 0.1.0

Description Combines jittered group spike-and-slab LASSO regularization with intermediate fusion to enable integrative learning across multiple data modalities. For uncertainty quantification, BayesCOOP applies the Bayesian bootstrap to generate approximate posterior samples by performing maximum a posteriori (MAP) estimation on jittered and resampled multimodal datasets. Currently, only continuous outcomes are supported.

License MIT + LICENSE

Encoding UTF-8

Imports caret, dplyr, glmnet, MCMCpack, rmutil, survival, truncnorm, utils

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Contents

`BayesCOOP` *BayesCOOP*

Description

This function implements the BayesCOOP methodology for supervised multimodal integration. It combines jittered group spike-and-slab LASSO regularization with intermediate fusion to enable integrative learning across multiple data modalities. For uncertainty quantification, BayesCOOP applies the Bayesian bootstrap to generate approximate posterior samples by performing maximum a posteriori (MAP) estimation on jittered and resampled multimodal datasets. Currently, only continuous outcomes are supported.

Usage

```
BayesCOOP(
  data_train,
  data_test,
  family = "gaussian",
  ss = c(0.05, 1),
  group = TRUE,
  bb = TRUE,
  alpha_dirich = 1,
  bbiters = 1100,
  bbburn = 100,
  maxit = 100,
  filter = TRUE,
  abd_thresh = 0,
  prev_thresh = 0.1,
  Warning = TRUE,
  verbose = TRUE,
  control = list()
)
```

Arguments

data_train	a list of feature_table, sample_metadata and feature_metadata from training data (unstandardized). See *IntegratedLearner* for more details.
data_test	a list of feature_table, sample_metadata and feature_metadata from testing data (unstandardized). See *IntegratedLearner* for more details.
family	currently supports only Gaussian family. Default: "gaussian".
ss	a length-2 numeric vector giving the spike/sLab scales c(s0, s1) with s0 < s1. Default: c(0.05, 1).
group	logical. If TRUE, predictors are grouped by modality and given a group spike-and-slab prior. If FALSE, no grouping is used. Default: TRUE.
bb	logical. If TRUE, run full Bayesian bootstrap inference; if FALSE, run MAP estimation over a user-supplied rho grid in control. Default: TRUE.
alpha_dirich	Dirichlet concentration for Bayesian bootstrap weights. Default: 1.
bbiters	number of Bayesian bootstrap iterations. Default: 1100.
bbburn	number of burn-in iterations discarded from the bootstrap. Default: 100.
maxit	maximum EM iterations in the inner optimizer. Default: 100.
filter	logical. If TRUE, apply abundance/prevalence filtering to features. Default: TRUE.
abd_thresh	minimum abundance threshold for keeping a feature. Default: 0.
prev_thresh	minimum prevalence threshold (proportion of samples above abd_thresh). Default: 0.1.
Warning	logical. If TRUE, emit non-convergence warnings. Default: TRUE.
verbose	logical. If TRUE, print iteration counts and runtime. Default: TRUE.
control	a named list with element rho, giving one or more candidate rho values to try when bb = FALSE.

Value

If `bb = TRUE`, a list with:

`y_samples`	posterior predictive draws (approximate)
`y_pred`	posterior predictive median for each held-out sample
`mspe`	mean squared prediction error on test data
`beta_samples`	posterior draws of regression coefficients
`beta_postmed`	posterior median coefficients
`rho_samples`	posterior draws of consensus penalty `rho`
`rho_postmed`	posterior median of `rho`
`errVar_samples`	posterior draws of residual variance
`time`	runtime in minutes

If `bb = FALSE`, a list with:

`y_pred`	predicted response values on test data
`mspe`	mean squared prediction error on test data
`beta_MAP`	MAP estimate of regression coefficients
`rho_MAP`	selected `rho` (minimizing MSPE over `control$rho`)
`time`	runtime in minutes

Index