Python Code Documentation

Function

`admissible_subsets`

Description

Performs variable selection in the linear model setting. Can to handle both $p \leq n$ and $p > n$ situations. See the reference for further details.

Usage

`admissible_subsets`(y, X, N, *steps*, *burnin*, *po*, $PropWeights$)

Arguments

y:	An $n \times 1$ NumPy array of response data.
X:	An $n \times p$ NumPy array of predictors.
N:	The number of importance samples used to estimate $\mathrm{E}(h(\beta_M))$ within the pseudo-marginal MCMC. Default is 100.
steps:	The number of MCMC steps.
burnin:	The number of initial *steps* to discard.
po:	Belief about the number of predictors in the true model. Can be chosen via cross-validation with `admissible_subsets_cv`.
PropWeights:	The weights used for proposing which predictors to add or drop as the MCMC samples index sets $M \subset \{1, \ldots, p\}$. Default is to use squared coefficient estimates from elastic net, via the `ElasticNetCV` function from the 'sklearn.linear_model' Python module, added by n^{-2}.

Values

chain:	A $(steps - burnin) \times p$ NumPy array containing the MCMC sample path (or trace) over index sets M, after *burnin* number of steps.
postSample:	A NumPy array containing the indices (in each row) for every M visited in the MCMC sample path.
postProbs:	A list containing the relative frequencies for which each of the index sets M in *postSample* was visited in the MCMC sample path.

AcceptRatio: The number of MCMC steps in which a proposed index set M was accepted.

References

J. P. Williams and J. Hannig (2017). Non-penalized variable selection in high-dimensional linear model settings via generalized fiducial inference. *Submitted.*

Supplement: `https://jonathanpw.github.io/assets/WilliamsHannig_supplement.pdf`

Function

`admissible_subsets_cv`

Description

Performs k-fold cross-validation to choose po in `admissible_subsets`, using an implementation of `admissible_subsets`. See the reference for further details.

Usage

`admissible_subsets_cv`(y, X, N, $steps$, $burnin$, $grid$, num_folds)

Arguments

y: An $n \times 1$ NumPy array of response data.

X: An $n \times p$ NumPy array of predictors.

N: The number of importance samples used to estimate $\mathrm{E}(h(\beta_M))$ within the pseudo-marginal MCMC, during the cross-validation procedure. Default is 30.

steps: The number of MCMC steps, during the cross-validation procedure. Default is 200.

burnin: The number of initial *steps* to discard, during the cross-validation procedure. Default is 100.

grid: A list of po values to perform cross-validation over. Default is $[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$.

num_folds: The number of folds to partition the data into for performing k-fold cross-validation. Default is 10.

Values

po: The optimal tuning parameter based on the implemented cross-validation.

References

J. P. Williams and J. Hannig (2017). Non-penalized variable selection in high-dimensional linear model settings via generalized fiducial inference. *Submitted*.

Supplement: `https://jonathanpw.github.io/assets/WilliamsHannig_supplement.pdf`