Package ‘GFLassoInference’

September 20, 2021

Title Powerful Graph Fused Lasso Inference

Version 1.0

Maintainer Yiqun T. Chen <yiqunc@uw.edu>

Author Yiqun Chen

Description An implementation of powerful selective inference for graph fused lasso as described in Chen et al. (2021+)

License GPL-3

Depends R (>= 3.5)

Imports Matrix, igraph, sparsesvd, MASS, Rmpfr, genlasso, intervals

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

Suggests rmarkdown,
knitr

VignetteBuilder knitr

R topics documented:

`fusedlasso_inf` *More powerful test for the graph fused lasso*

Description

This functions tests the null hypothesis of no difference in means between connected components `c1` and `c2` of the output of the graph fused lasso solution. The ordering are numbered as per the results of the `fusedlasso` function in the `genlasso` package.

Usage

```
fusedlasso_inf(
  y,
  D,
  c1,
  c2,
  method,
  sigma,
  K = NULL,
  L = NULL,
  early_stop = NULL,
  compute_ci = FALSE,
  alpha_level = 0.05
)
```

Arguments

`y`	Numeric vector; n dimensional observed data
`D`	Numeric matrix; m by n penalty matrix, i.e., the oriented incidence matrix over the underlying graph
`c1, c2`	Integers selecting the two connected components to test, as indexed by the results of `genlasso::fusedlasso`.
`method`	One of "K" or "CC", which indicates which conditioning set to use
`sigma`	Numeric; noise standard deviation for the observed data, a non-negative number.
`K`	Integer; number of steps to run the dual-path algorithm. It must be specified if method=="K".
`L`	Integer; the targeted number of connected components. It must be specified if method=="CC".
`early_stop`	Numeric; specify when the truncation set computation should be terminated. The default is NULL, which indicates infinity.
`compute_ci`	Logical; the default is False. Specifying whether confidence intervals for $\nu^T\beta$, the difference in means between the two estimated connected components, should be computed.
`alpha_level`	Numeric; parameter for the 1-`alpha_level` confidence interval, defeault to 0.05

Details

Currently, we support two different conditioning sets: conditioning set 1 is based on the output after K steps dual-path algorithm; and conditioning set 2 is based on the output of the after the dual-path algorithm yields c connected components in the output.

Input:

Consider the generative model $Y_j = \beta_j + \epsilon_j, \epsilon_j \sim N(0, \sigma^2). j = 1, ..., n$, where the underlying signal β is assumed to be piecewise constant with respect to an underlying graph. The fused lasso estimate minimizes the following objective function

$$minimize_\beta \frac{1}{2} \sum_{j=1}^{n} (y_j - \beta_j)^2 + \lambda \sum_{(i,j)\in E} |\beta_i - \beta_j|,$$

where E is the edge set of the underlying graph. The solution $\hat{\beta}$ can then be segment into connected components; that is, the set of $\hat{\beta}$ that takes on the same value, and are connected in the original graph.

Now suppose we want to test whether the means of two estimated connected components `c1` and `c2` are equal; or equivalently, the null hypothesis of the form $H_0 : \nu^T \beta = 0$ versus $H_1 : \nu^T \beta \neq 0$ for suitably chosen ν.

This function computes the following p-value:

$$P(|\nu^T Y| \geq |\nu^T y| \mid \hat{C}_1, \hat{C}_2 \in CC_K(Y), \Pi_\nu^\perp Y = \Pi_\nu^\perp y)$$

, where $CC_K(Y)$ is the set of estimated connected components from applying K steps of the dual path algorithm on data Y , and $\Pi_\nu^\perp$ is the orthogonal projection to the orthogonal complement of ν. In particular, the test based on this p-value controls the selective Type I error and has higher power than an existing method by Hyun et al. (2018). Readers can refer to the Section 3 in Chen et al. (2021+) for more details.

Value

Returns a list with elements:

- `Union` the p-value proposed in Chen et al. (2021+)
- `truncation_set` the conditioning set of Chen et al. (2021+) stored as `Intervals` class
- `test_stats` test statistics: the difference in means of two connected components
- `beta_hat` Graph fused lasso estimates
- `connected_comp` Estimated connected component
- `Naive` the naive p-value using a z-test
- `Hyun` the p-value proposed in Hyun et al. (2018)
- `hyun_set` the conditioning set of Hyun et al. (2018) stored as `Intervals` class
- `CI_result` confidence interval of level 1-`alpha_level` if `compute_ci=TRUE`

References

Chen YT, Jewell SW, Witten DM. (2021+) More powerful selective inference for the graph fused lasso Hyun S, G'Sell M, Tibshirani RJ. (2018) Exact post-selection inference for the generalized lasso path. Electron J Stat.

Examples

```
lev1 <- 0 # mean for group 1
lev2 <- 3 # mean (absolute value) for group 2/3
sigma <- 1 # level of noise
nn <- 8 # grid size
Dmat <- genlasso::getD2d(nn, nn) # generate D matrix for the 2D fused lasso
### Create the underlying signal
A <- matrix(lev1, ncol=nn, nrow = nn)
A[1:round(nn/3),1:round(nn/3)] <- 1*lev2
A[(nn-2):(nn),(nn-2):(nn)] <- -1*lev2
### Visualize the underlying signal
lattice::levelplot(A)
set.seed(2005)
A.noisy <- A + rnorm(nn^2,mean=0,sd=sigma)
y <- c(t(A.noisy))
```

```
### Now use the fusedlasso function to obtain estimated connected components after K=13
### steps of the dual path algorithm
K = 13
complete_sol <- genlasso::fusedlasso(y=y,D=Dmat,maxsteps=K)
beta_hat <- complete_sol$beta[,K]
### estimated connected components
estimated_CC <- complete_sol$pathobjs$i
estimated_CC
### Run a test for a difference in means between estimated connected components 1 and 2
result_demo <- fusedlasso_inf(y=y, D=Dmat, c1=1, c2=2, method="K", sigma=sigma, K=K)
summary(result_demo)
```

`GFLassoInference` *Quantifying uncertainty for spikes estimated from calcium imaging data*

Description

GFLassoInference is an R packagefor testing for a difference in means between a pair of connected components estimated from the graph fused lasso. Details can be found in our manuscript (Chen et al. 2021+). In addition to the manual in R, a detailed tutorial can be found at https://yiqunchen.github.io/GFLassoInferenc

Details

Maintainer: Yiqun T. Chen <yiqunc@uw.edu> The main function in the package is `fusedlasso_inf`. In our tutorials and experiments, we also used functions from the `genlasso` package and the `genlassoinf` package.

References

Chen YT, Jewell SW, Witten DM. (2021+) More powerful selective inference for the graph fused lasso

Hyun S, G'Sell M, Tibshirani RJ. (2018) Exact post-selection inference for the generalized lasso path. Electron J Stat.

Tibshirani RJ, Taylor J. The solution path of the generalized lasso. Ann Stat. 2011;39(3):1335-1371. doi:10.1214/11-AOS878

Examples

```
lev1 <- 0 # mean for group 1
lev2 <- 3 # mean (absolute value) for group 2/3
sigma <- 1 # level of noise
nn <- 8 # grid size
Dmat <- genlasso::getD2d(nn, nn) # generate D matrix for the 2D fused lasso
### Create the underlying signal
A <- matrix(lev1, ncol=nn, nrow = nn)
A[1:round(nn/3),1:round(nn/3)] <- 1*lev2
A[(nn-2):(nn),(nn-2):(nn)] <- -1*lev2
### Visualize the underlying signal
lattice::levelplot(A)
set.seed(2005)
A.noisy <- A + rnorm(nn^2,mean=0,sd=sigma)
```

```
y <- c(t(A.noisy))
### Now use the fusedlasso function to obtain estimated connected components after K=13
### steps of the dual path algorithm
K = 13
complete_sol <- genlasso::fusedlasso(y=y,D=Dmat,maxsteps=K)
beta_hat <- complete_sol$beta[,K]
### estimated connected components
estimated_CC <- complete_sol$pathobjs$i
estimated_CC
### Run a test for a difference in means between estimated connected components 1 and 2
result_demo <- fusedlasso_inf(y=y, D=Dmat, c1=1, c2=2, method="K", sigma=sigma, K=K)
summary(result_demo)
```

---

summary.fusedlasso_inf

Summarize the result for powerful graph fused lasso inference

---

Description

Summarize the result for powerful graph fused lasso inference

Usage

```
summary.fusedlasso_inf(object, ...)
```

Arguments

object	output from running spike_estimates
...	to be passed to methods

Value

A data frame with summarized results

Examples

```
lev1 <- 0 # mean for group 1
lev2 <- 3 # mean (absolute value) for group 2/3
sigma <- 1 # level of noise
nn <- 8 # grid size
Dmat <- genlasso::getD2d(nn, nn) # generate D matrix for the 2D fused lasso
### Create the underlying signal
A <- matrix(lev1, ncol=nn, nrow = nn)
A[1:round(nn/3),1:round(nn/3)] <- 1*lev2
A[(nn-2):(nn),(nn-2):(nn)] <- -1*lev2
### Visualize the underlying signal
lattice::levelplot(A)
set.seed(2005)
A.noisy <- A + rnorm(nn^2,mean=0,sd=sigma)
y <- c(t(A.noisy))
### Now use the fusedlasso function to obtain estimated connected components after K=13
### steps of the dual path algorithm
K = 13
complete_sol <- genlasso::fusedlasso(y=y,D=Dmat,maxsteps=K)
```

```
beta_hat <- complete_sol$beta[,K]
### estimated connected components
estimated_CC <- complete_sol$pathobjs$i
estimated_CC
### Run a test for a difference in means between estimated connected components 1 and 2
result_demo <- fusedlasso_inf(y=y, D=Dmat, c1=1, c2=2, method="K",
sigma=sigma, K=K, compute_ci=TRUE)
summary(result_demo)
```

Index