Package ‘LPRelevance’

May 17, 2020

Type Package

Title Relevance-Integrated Statistical Inference Engine

Version 3.1

Date 2020-05-15

Author Subhadeep Mukhopadhyay, Kaijun Wang

Maintainer Kaijun Wang <kaijunwang.19@gmail.com>

Description A framework of methods to perform customized inference at individual level by taking contextual covariates into account. Three main functions are provided in this package: (i) LASER(): it generates specially-designed artificial relevant samples for a given case; (ii) g2l.proc(): computes customized fdr(z|x); and (iii) rEB.proc(): performs empirical Bayes inference based on LASERs. The details can be found in Mukhopadhyay, S., and Wang, K (2020, Technical Report).

Imports leaps,locfdr,Bolstad2,reshape2,ggplot2,polynom,glmnet,caret

Depends R (>= 3.5.0), stats, BayesGOF, MASS

License GPL-2

NeedsCompilation no

Repository CRAN

Date/Publication 2020-05-16 22:30:10 UTC

R topics documented:

`LPRelevance-package` *Relevance-Integrated Statistical Inference Engine*

Description

How to individualize a global inference model? The goal of this package is to provide a systematic recipe for converting a classical global inference algorithm into a customized one. It provides methods that perform individual level inferences by taking contextually relevant covariates into account. At the heart of our solution is the concept of "artificially-designed relevant samples", called LASERs–which pave the way to construct an inference mechanism that is simultaneously efficiently estimable and contextually relevant, thus works at both macroscopic (overall simultaneous) and microscopic (individual-level) scale.

Author(s)

Subhadeep Mukhopadhyay, Kaijun Wang

Maintainer: Kaijun Wang <kaijunwang.19@gmail.com>

References

Mukhopadhyay, S., and Wang, K (2020) "On The Problem of Relevance in Statistical Inference". Technical Report.

`data.dti` *DTI data.*

Description

A diffusion tensor imaging study comparing brain activity of six dyslexic children versus six normal controls. Two-sample tests produced z-values at $N = 15443$ voxels (3-dimensional brain locations), with each $z_i \sim N(0, 1)$ under the null hypothesis of no difference between the dyslexic and normal children.

Usage

```
data(data.dti)
```

Format

A data frame with 15443 observations on the following 4 variables.

`coordx` A list of x coordinates

`coordy` A list of y coordinates

`coordz` A list of z coordinates

`z` The z-values.

Source

http://statweb.stanford.edu/~ckirby/brad/LSI/datasets-and-programs/datasets.html

References

Efron, B. (2012). "Large-scale inference: empirical Bayes methods for estimation, testing, and prediction". Cambridge University Press.

`funnel` *Simulation data set.*

Description

A simulated heterogeneous data set used in our paper.

Usage

```
data("funnel")
```

Format

A data frame with 3565 observations on the following 3 variables.

`x` A list of covariate values.

`z` A list of z-values.

`tags` Binary vector of labels, 1 indicates a data point is a signal.

References

Mukhopadhyay, S., and Wang, K (2020) "On The Problem of Relevance in Statistical Inference". Technical Report.

`g2l.proc` *Procedures for global and local inference.*

Description

This function performs customized fdr analyses tailored to each individual cases.

Usage

```
g2l.proc(X, z, X.target = NULL, z.target = NULL, m = c(4, 6), alpha = 0.05,
niter = NULL, nsample = length(z), lp.reg.method = "lm",
null.scale = "QQ", approx.method = "direct", ngrid = 2000,
centering = TRUE, coef.smooth = "BIC", fdr.method = "locfdr",
plot = TRUE, rel.null = "custom", locfdr.df = 10,
fdr.th.fixed = NULL, parallel = TRUE, ...)
```

Arguments

`X` A n-by-d matrix of covariate values

`z` A length n vector containing observations of z values.

`X.target` A k-by-d matrix providing k sets of covariates for target cases to investigate. Set to NULL to investigate all cases and provide global inference results.

`z.target` A vector of length k, providing the target z values to investigate

`m` An ordered pair. First number indicates how many LP-nonparametric basis to construct for each X, second number indicates how many to construct for z. Default: `m=c(4,6)`.

`alpha` Confidence level for determining signals.

`niter` Number of iterations to use for each target case, each time a new set of relevance samples will be generated for analysis, and the resulting fdr curves are aggregated together by taking the mean values. Set to `NULL` to disable.

`nsample` Number of relevance samples generated for each case. The default is the size of the input z-statistic.

`lp.reg.method` Method for estimating the relevance function and its conditional LP-Fourier coefficients. We currently support three options: lm (inbuilt with subset selection), glmnet, and knn.

`null.scale` Method of estimating null standard deviation from the laser samples. Available options: "IQR", "QQ" and "locfdr"

`approx.method` Method used to approximate customized fdr curve, default is "direct".When set to "indirect", the customized fdr is computed by modifying pooled fdr using relevant density function.

`ngrid` Number of gridpoints to use for computing customized fdr curve.

`centering` Whether to perform regression-adjustment to center the data, default is TRUE.

`coef.smooth` Specifies the method to use for LP coefficient smoothing (AIC or BIC). Uses BIC by default.

`fdr.method` Method for controlling false discoveries (either "locfdr" or "BH"), default choice is "locfdr".

`plot` Whether to include plots in the results, default is `TRUE`.

`rel.null` How the relevant null changes with x: "custom" denotes we allow it to vary with x, and "th" denotes fixed.

`locfdr.df` Degrees of freedom to use for `locfdr()`

`fdr.th.fixed` Use fixed fdr threshold for finding signals. Default set to `NULL`, which finds different thresholds for different cases.

`parallel` Use parallel computing for obtaining the relevance samples, mainly used for very huge `nsample`, default is FALSE.

`...` Extra parameters to pass to other functions. Currently only supports the arguments for `knn()`.

Value

A list containing the following items:

`macro` Available when `X.target` set to `NULL`, contains the following items:

`result` A list of global inference results:

`X` Matrix of covariates, same as input `X`.

`z` Vector of observations, same as input `z`.

`probnull` A vector of length n, indicating how likely the observed z belongs to local null.

`signal` A binary vector of length n, discoveries are indicated by 1.

`plots` A list of plots for global inference:

`signal_x` A plot of signals discovered, marked in red

`dps_xz` A scatterplot of z on x, colored based on the discovery propensity scores, only available when `fdr.method = "locfdr"`.

`dps_x` A scatterplot of discovery propensity scores on x, only available when `fdr.method = "locfdr"`.

`micro` Available when `X.target` are provided with values, contains the following items:

`result` Customized estimates for null probabilities for target X and z

`global` Pooled global estimates for null probabilities for target X and z

`plots` Customized fdr plots for the target cases.

`m.lp` Same as input `m`

Author(s)

Subhadeep Mukhopadhyay, Kaijun Wang

Maintainer: Kaijun Wang <kaijunwang.19@gmail.com>

References

Mukhopadhyay, S., and Wang, K (2020) "On The Problem of Relevance in Statistical Inference". Technical Report.

Examples

```
data(funnel)
X<-funnel$x
z<-funnel$z
##macro-inference using locfdr and LASER:
g2l_macro<-g2l.proc(X,z,m=c(4,8),niter=NULL,alpha=.05,
fdr.method = 'locfdr',parallel=FALSE)
g2l_macro$macro$plots
```

```
##micro-inference on point (30,4.09), using 10 iterations:
X.target=30
z.target=4.09
g2l_micro<-g2l.proc(X,z,X.target,z.target,niter = 10,m=c(4,8),alpha=.05,parallel=FALSE)
g2l_micro$micro$result
g2l_micro$micro$global
g2l_micro$micro$plots
```

kidney *Kidney data.*

Description

This data set records age and kidney function of $N = 157$ volunteers. Higher scores indicates better function.

Usage

```
data(kidney)
```

Format

A data frame with 157 observations on the following 2 variables.

- `x` A list of patients' age.
- `z` A list of kidney scores.

Source

http://statweb.stanford.edu/~ckirby/brad/LSI/datasets-and-programs/datasets.html

References

Efron, B. (2012). "Large-scale inference: empirical Bayes methods for estimation, testing, and prediction". Cambridge University Press.

Lemley, K. V., Lafayette, R. A., Derby, G., Blouch, K. L., Anderson, L., Efron, B., & Myers, B. D. (2007). "Prediction of early progression in recently diagnosed IgA nephropathy." Nephrology Dialysis Transplantation, 23(1), 213-222.

LASER *Generates Artificial RELevance Samples.*

Description

This function generates the artificial relevance samples (LASER). These are "sharpened" z-samples manufactured by the relevance-function $d_{Y|X=x}(u)$.

Usage

```
LASER( X,z, X.target, m=c(4,6), nsample=length(z), lp.reg.method='lm',
       coef.smooth='BIC', centering=TRUE,parallel=FALSE,...)
```

Arguments

`X` A n-by-d matrix of covariate values

`z` A length n vector containing observations of z values.

`X.target` A k-by-d matrix providing k sets of target points for which the LASERs are required.

`m` An ordered pair. First number indicates how many LP-nonparametric basis to construct for each X, second number indicates how many to construct for z. Default: `m=c(4,6)`

`nsample` Number of relevance samples to generate for each case.

`lp.reg.method` Method for estimating the relevance function and its conditional LP-Fourier coefficients. We currently support thee options: lm (inbuilt with subset selection), glmnet, and knn.

`centering` Whether to perform regression-adjustment to center the data, default is TRUE.

`coef.smooth` Specifies the method to use for LP coefficient smoothing (AIC or BIC). Uses BIC by default.

`parallel` Use parallel computing for obtaining the relevance samples, mainly used for very huge `nsample`, default if FALSE.

`...` Extra parameters to pass to other functions. Currently only supports the arguments for `knn()`.

Value

A list containing the following items:

`data` The relevance sample points generated for `X.target`.

`LPcoef` The LP coefficient values for z given $\boldsymbol{x}$.

Author(s)

Subhadeep Mukhopadhyay, Kaijun Wang

Maintainer: Kaijun Wang <kaijunwang.19@gmail.com>

References

Mukhopadhyay, S., and Wang, K (2020) "On The Problem of Relevance in Statistical Inference". Technical Report.

Examples

```
data(funnel)
X<-funnel$x
z<-funnel$stat
z.laser.x30<-LASER(X,z,X.target=30,m=c(4,8))$data
hist(z.laser.x30,50)
```

---

rEB.proc *Relevance-Integrated Empirical Bayes Inference*

Description

Performs custom-tailored empirical Bayes inference via LASERs.

Usage

```
rEB.proc(X, z, X.target, z.target, m = c(4, 6), niter = NULL, centering = TRUE,
lp.reg.method = "lm", coef.smooth = "BIC", nsample = length(z),
theta.set.prior = NULL, theta.set.post = NULL, LP.type = "L2",
g.method = "DL", sd0 = NULL, m.EB = 8, parallel = FALSE,
avg.method = "mean", post.curve = "HPD", post.alpha = 0.8,
color = "red", ...)
```

Arguments

`X`	A n-by-d matrix of covariate values
`z`	A length n vector containing observations of target random variable.
`X.target`	A length d vector providing the set of covariates for the target case.
`z.target`	the target z to investigate
`m`	An ordered pair. First number indicates how many LP-nonparametric basis to construct for each X, second number indicates how many to construct for z.
`niter`	Number of iterations to use for Finite Bayes, set to `NULL` to disable.
`centering`	Whether to perform regression-adjustment to center the data, default is TRUE.
`lp.reg.method`	Method for estimating the relevance function and its conditional LP-Fourier coefficients. We currently support thee options: lm (inbuilt with subset selection), glmnet, and knn.
`coef.smooth`	Specifies the method to use for LP coefficient smoothing (AIC or BIC). Uses BIC by default.

`nsample` Number of relevance samples generated for the target case.

`theta.set.prior` This indicates the set of grid points to compute prior density.

`theta.set.post` This indicates the set of grid points to compute posterior density.

`LP.type` User selects either "L2" for LP-orthogonal series representation of relevance density function d or "MaxEnt" for the maximum entropy representation. Default is L2.

`g.method` Determines the method to find τ^2: "DL" uses Dersimonian and Lard technique,"SJ" uses Sidik-Jonkman

`sd0` Fixed standard error for $z|\theta$. Default is NULL, the standard error will be calculated from data.

`m.EB` The truncation point reflecting the concentration of true nonparametric prior density π around known prior distribution g

`parallel` Use parallel computing for obtaining the relevance samples, mainly used for very huge `nsample`, default if FALSE.

`avg.method` For Finite Bayes, this specifies how the results from different iterations are aggregated. ("`mean`" or "`median`".)

`post.curve` For plotting, this specifies what to show on posterior curve. "`HPD`" provides HPD interval, "band" gives confidence band.

`post.alpha` Confidence level to use when plotting posterior confidence band, or the alpha level for HPD interval.

`color` The color of the plots.

`...` Extra parameters to pass to other functions. Currently only supports the arguments for `knn()`.

Value

A list containing the following items:

`result` contains the results for prior and posterior density:

`prior` Prior results:

`g.par` Parameters for g.

`LP.coef` reports the LP coefficient values for z given $\boldsymbol{x}$.

`posterior` Posterior results:

`post.mean` Posterior mean for $\pi(\theta|\boldsymbol{x})$.

`post.mean.sd` Standard error for the posterior mean. Only available for Finite Bayes.

`HPD.interval` The HPD interval for posterior $\pi(\theta|\boldsymbol{x})$.

`post.alpha` same as input `post.alpha`.

`plots` The plots for prior and posterior density.

Author(s)

Subhadeep Mukhopadhyay, Kaijun Wang

Maintainer: Kaijun Wang <kaijunwang.19@gmail.com>

References

Mukhopadhyay, S., and Wang, K (2020) "On The Problem of Relevance in Statistical Inference". Technical Report.

Examples

```
data(funnel)
X<-funnel$x
z<-funnel$stat
X.target=30
z.target=4.09
rEB.out<-rEB.proc(X,z,X.target,z.target,m=c(4,8),
theta.set.prior=seq(-2,2,length.out=200),
theta.set.post=seq(-2,5,length.out=200),
centering=TRUE,m.EB=6,parallel=FALSE)
rEB.out$plots$rEB.post
rEB.out$plots$rEB.prior
```

Index