Package 'graphseg'

October 4, 2023

Type Package

Title Segmentation of Graph-Based Signals

Version 0.1.1

Maintainer Vivien Goepp <vivien.goepp@gmail.com>

Description Perform segmentation of graph-based signals. Assume a noisy observation of a signal two values correspond to vertices on a graph. Assume the true value of the signal is piece-wise constant (where each 'piece' is a connected subgraph). The main function, agraph(), computes the segmentation of the signal. The package also includes a wrapper around the competing method flsa() (from package 'flsa').
More information about this method in Goepp and van de Kassteele (2022) ``Graph-Based Spatial Segmentation of Health-Related Areal Data'' <doi:10.48550/arXiv.2206.06752>.

License GPL (>= 3)

Encoding UTF-8

LazyData true

URL https://github.com/goepp/graphseg

Imports igraph, magrittr, methods, Matrix, sf, flsa

RoxygenNote 7.1.1

Depends R (>= 2.10)

Suggests testthat, knitr, rmarkdown

NeedsCompilation no

Author Vivien Goepp [aut, cre] (<https://orcid.org/0000-0001-6961-4260>)

Repository CRAN

Date/Publication 2023-10-03 22:10:02 UTC

R topics documented:

---

agraph *Segmentation using graph structure*

Description

These functions provide a clustering of a signal on graph into a piecewise constant signal on graph. Given a graph and a signal gamma assigning a value to each node, it returns another signal which is constant over subgraphs where gamma has close to equal value. See references.

Only parameters gamma and graph need be provided. The other parameters concern the internals of the estimating procedure and usually do not need to be changed. agraph is the general-purpose function. agraph_prec does the same thing as agraph in the case where gamma as a covariance structure. It is provided as the precision matrix prec, which has to be a sparse matrix (Matrix::sparseMatrix) for fast computation. See Goepp and van de Kassteele (2021).

Usage

```
agraph(
  gamma,
  graph,
  lambda = 10^seq(-4, 4, length.out = 50),
  weights = NULL,
  shrinkage = TRUE,
  delta = 1e-10,
  tol = 1e-08,
  thresh = 0.01,
  itermax = 50000
)

agraph_prec(
  gamma,
  graph,
  prec,
  lambda = 10^seq(-4, 4, length.out = 50),
  weights = NULL,
  shrinkage = TRUE,
  delta = 1e-10,
  tol = 1e-08,
  thresh = 0.01,
  itermax = 10000
)
```

Arguments

gamma	input vector to regularize
graph	an igraph object (from package igraph) giving the regularization structure
lambda	regularizing constant
weights	weights for gamma. Default value is one.
shrinkage	Boolean, defaults TRUE. Whether to return the adaptive ridge estimate as output. If FALSE, the adaptive ridge is used to define a segmentation into zones, and the signal is estimated on each zone using non-penalized estimation.
delta	Computational constant in the adaptive ridge reweighting formula.
tol	Tolerance to test for convergence of the adaptive ridge
thresh	Thresholding constant used to fuse two adjacent regions with close value of gamma.
itermax	Total number of iterations. Default value is 10000. Setting a low value can make the procedure return NULL entries for some values of lambda.
prec	precision matrix (inverse of the variance-covariance matrix). Has to be a sparse matrix for efficiency.

Value

A list with the following elements:

- result: matrix whose rows are the segmented output of input signal gamma, for each value of lambda
- bic, gcv, and aic: vectors of length length(lambda), giving the BIC, GCV, and AIC criteria for each value of lambda. See references below.
- model_dim, nll: vectors of length length(lambda), giving the model dimension and negative log-likelihood for each value of lambda. See reference below for the definition of these terms.

References

Schwarz G. (1978) Estimating the Dimension of a Model. Ann. Statist. 6 (2) 461 - 464, March, 1978. doi:10.1214/aos/1176344136

Akaike H. (1974) A new look at the statistical model identification, in IEEE Transactions on Automatic Control, vol. 19, no. 6, pp. 716-723, December 1974 doi:10.1109/TAC.1974.1100705

Hastie T., Friedman J., and Tibshirani R. (2009) The elements of statistical learning: data mining, inference, and prediction (Vol. 2, pp. 1-758). New York: Springer doi:10.1007/9780387216065

Goepp V. and van de Kassteele J. (2021) Graph-Based Spatial Segmentation of Health-Related Areal Data, arxiv preprint. doi:10.48550/arXiv.2206.06752

See Also

flsa_graph()

agraph_one_lambda *Segmentation using graph structure*

Description

Segmentation using graph structure

Usage

```
agraph_one_lambda(
  gamma,
  graph,
  lambda = 1,
  weights = NULL,
  delta = 1e-10,
  tol = 1e-08,
  thresh = 0.01
)
```

Arguments

gamma	entry vector to regularize
graph	an igraph object (from package igraph) giving the regularization structure
lambda	regularizing constant
weights	weights for gamma. Default value is one.
delta	Computational constant in the adaptive ridge reweighting formula.
tol	Tolerance to test for convergence of the adaptive ridge
thresh	Thresholding constant used to fuse two adjacent regions with close value of gamma.

departement *French departement*

Description

Geographical data of the French administrative units "département" This data set excludes overseas departement as well as the two departements of the island of Corsica.

Usage

```
departement
```

Format

An sf object, with lattitude and longitude

Source

The data set comes from https://github.com/gregoiredavid/france-geojson/, under the "Licence Ouverte / Open Licence v2.0" licence.

Examples

```
data(departement)
plot(departement["geometry"])
```

---

flsa_graph *Segmentation using graph structure and the fused lasso estimate*

Description

Wrapper around the function `flsa::flsa`, which computes the fused lasso signal approximator (see reference). Like `agraph`, this function takes a signal on graph and returns a clustering thereof into a piecewise-constant signal. The difference with `agraph` is the estimation method: `agraph` works well when the true signal is sparse and its computation time scales well to large graphs.

Usage

```
flsa_graph(gamma, graph, lambda)
```

Arguments

`gamma`	entry vector to regularize
`graph`	graph (an igraph object) giving the regularization structure
`lambda`	regularizing constant

Value

A list with the following elements:

- `result`: matrix whose rows are the segmented output of input signal `gamma`, for each value of `lambda`
- `bic`, `gcv`, and `aic`: vectors of length `length(lambda)`, giving the BIC, GCV, and AIC criteria for each value of lambda. See references below.
- `model_dim`, `nll`: vectors of length `length(lambda)`, giving the model dimension and negative log-likelihood for each value of lambda. See reference below for the definition of these terms.

References

Hoefling, H., A Path Algorithm for the Fused Lasso Signal Approximator, Journal of Computational and Graphical Statistics (2010) doi:10.1198/jcgs.2010.09208

See Also

graphseg::agraph()

`paris` *IRIS of Paris*

Description

Geographical data of the French statistical units "IRIS" (*grouped islets for statistical information*) forming the city of Paris. Five iris units (forming two parcs: that of Boulogne and Vincennes) of dispropotionate size have been removed from the data set.

Usage

```
paris
```

Format

An sf object, with lattitude and longitude

Source

The data set comes from the French National Statistics Institute and the National Geography Institute. It is accessible at `https://www.data.gouv.fr/fr/datasets/contours-iris-insee-ign/`, under the "Licence Ouverte / Open Licence v2.0" licence.

Examples

```
data(paris)
plot(paris["geometry"])
```

utrecht_district *Administrative areas of the Netherlands aroung the city of Utrecht*

Description

Geographical data of the Dutch administrative units "District".

Usage

```
utrecht_district

graph_utrecht_district
```

Format

utrecht_district An sf object, with lattitude and longitude. With 7 variables and 650 zones.

graph_utrecht_district The adjacency graph as an igraph object.

An object of class sf (inherits from data.frame) with 650 rows and 7 columns.

An object of class igraph of length 650.

Source

The data set comes from https://geodata.nationaalgeoregister.nl/.

Examples

```
data(utrecht_district); data(graph_utrecht_district)
coord <- sf::st_coordinates(sf::st_centroid(utrecht_district))
adj_municip <- as(as(igraph::as_adjacency_matrix(graph_utrecht_district, type = "both"),
                     "symmetricMatrix"),
                  "TsparseMatrix")
edge_list <- data.frame(adj_municip@i + 1, adj_municip@j + 1)
segment_df <- cbind(coord[edge_list[, 1], ], coord[edge_list[, 2], ])
ptmat <- as.matrix(segment_df[, 1:4])[2:nrow(segment_df), ]
linesegs <- lapply(split(ptmat, 1:nrow(ptmat)), function(x) {
    x <- matrix(x, nrow = 2, byrow = TRUE)
    x <- sf::st_linestring(x)})
final_sf <- sf::st_sf(sf::st_sfc(linesegs), 'ID' = 1:length(sf::st_sfc(linesegs)))
op <- par(mar = rep(0, 4))
plot(sf::st_geometry(utrecht_district), lwd = 0.6, border = "grey")
plot(sf::st_geometry(final_sf), lwd = 0.5, add = TRUE)
plot(sf::st_centroid(utrecht_district), add = TRUE, col = "black", pch = 20,
     cex = 0.5)
par(op)
```

Index