Package 'ivgls'

June 24, 2026

Type Package

Title Network-Aware IV Regression with Graph-Fused Lasso

Version 0.1.0

Description Implements network-aware instrumental variable regression for causal node discovery in high-dimensional settings with graph-structured exposures. Provides IVGL and IVGL-S estimators combining graph-Laplacian penalization with IV-based identification, including correction for invalid instruments via a sisVIVE-style update. Methods are described in Pal and Ghosh (2026) <doi:10.48550/arXiv.2604.24969>. The 'glmgraph' package, required for the main estimators, is available at the additional repository <https://djghosh1123.r-universe.dev>.

License MIT + file LICENSE

Encoding UTF-8

Imports glmnet, MASS, igraph

Suggests knitr, glmgraph, rmarkdown, testthat (>= 3.0.0), ggplot2, spelling

Additional_repositories https://djghosh1123.r-universe.dev

VignetteBuilder knitr

RoxygenNote 7.3.2

Config/roxygen2/version 8.0.0

Config/testthat/edition 3

URL https://github.com/djghosh1123/ivgls

BugReports https://github.com/djghosh1123/ivgls/issues

Language en-US

NeedsCompilation no

Author Dhrubajyoti Ghosh [aut, cre] (ORCID: <https://orcid.org/0000-0002-3360-3786>), Samhita Pal [aut] (ORCID: <https://orcid.org/0009-0001-4930-916X>)

Maintainer Dhrubajyoti Ghosh <dghosh3@kennesaw.edu>

Repository CRAN

Date/Publication 2026-06-24 09:30:02 UTC

Contents

`corrupt_graph` *Corrupt a graph by random edge swaps*

Description

Corrupt a graph by random edge swaps

Usage

```
corrupt_graph(A, corruption_rate = 0.3)
```

Arguments

`A`	Symmetric p x p binary adjacency matrix.
`corruption_rate`	Proportion of edges to remove and replace.

Value

A corrupted adjacency matrix.

eval_support *Compute performance metrics for support recovery*

Description

Compute performance metrics for support recovery

Usage

```
eval_support(true_support, estimated_support, p)
```

Arguments

true_support	Integer vector of true active indices.
estimated_support	Integer vector of estimated active indices.
p	Total number of predictors.

Value

Named numeric vector with MCC, TPR, FPR, and Selected.

generate_beta *Generate a sparse true coefficient vector on a graph*

Description

Generate a sparse true coefficient vector on a graph

Usage

```
generate_beta(
  A,
  s2 = 5,
  signal = 3,
  pattern = c("smooth", "nonsmooth", "community"),
  smooth_noise = 0.2
)
```

Arguments

A	Symmetric p x p adjacency matrix.
s2	Number of active nodes.
signal	Causal effect magnitude.
pattern	One of "smooth", "nonsmooth", or "community".
smooth_noise	SD of noise added around the base signal.

Value

A list with `beta_true` and `active_set`.

`generate_data` *Simulate data for graph-IV regression*

Description

Simulate data for graph-IV regression

Usage

```
generate_data(
  n = 100,
  p = 70,
  q = 500,
  s1 = 0.1,
  s_alpha = 10,
  alpha_strength = 5,
  beta_true
)
```

Arguments

`n`	Sample size.
`p`	Number of exposures.
`q`	Number of instruments.
`s1`	Fraction of instruments relevant for each exposure.
`s_alpha`	Number of invalid instruments.
`alpha_strength`	Direct-effect magnitude of invalid instruments.
`beta_true`	Numeric vector of length p of true causal effects.

Value

A list with `Y`, `X`, `Z`, `A_true`, and `alpha_true`.

get_laplacian *Compute the unnormalised graph Laplacian*

Description

Compute the unnormalised graph Laplacian

Usage

```
get_laplacian(A)
```

Arguments

A	Symmetric p x p binary adjacency matrix.

Value

A p x p Laplacian matrix.

get_mcc *Matthews Correlation Coefficient for support recovery*

Description

Matthews Correlation Coefficient for support recovery

Usage

```
get_mcc(true_support, estimated_support, p)
```

Arguments

true_support	Integer vector of truly active indices.
estimated_support	Integer vector of estimated active indices.
p	Total number of predictors.

Value

A scalar between -1 and 1.

ivgl *IVGL: IV regression with graph-fused Lasso*

Description

IVGL: IV regression with graph-fused Lasso

Usage

```
ivgl(Y, X, Z, L)
```

Arguments

Y	Numeric vector of length n. Outcome.
X	Numeric n x p matrix of endogenous exposures.
Z	Numeric n x q matrix of instruments.
L	Numeric p x p graph Laplacian (see get_laplacian).

Value

Numeric vector of length p of estimated causal effects.

ivgl_s *IVGL-S: IV regression with graph Lasso and invalid-IV correction*

Description

Extends IVGL with an alternating sisVIVE-style update to handle partially invalid instruments that violate the exclusion restriction.

Usage

```
ivgl_s(Y, X, Z, L, max_iter = 20, verbose = FALSE)
```

Arguments

Y	Numeric vector of length n. Outcome.
X	Numeric n x p matrix of endogenous exposures.
Z	Numeric n x q matrix of instruments.
L	Numeric p x p graph Laplacian.
max_iter	Maximum number of alternating iterations. Default 20.
verbose	Print CV loss at each iteration. Default FALSE.

Value

A list with beta (length p causal effects) and alpha (length q direct IV-outcome effects).

iv_lasso *IV-LASSO: Two-stage LASSO without graph structure*

Description

IV-LASSO: Two-stage LASSO without graph structure

Usage

```
iv_lasso(Y, X, Z)
```

Arguments

Y	Numeric vector of length n. Outcome.
X	Numeric n x p matrix of endogenous exposures.
Z	Numeric n x q matrix of instruments.

Value

Numeric vector of length p of estimated causal effects.

make_graph *Construct a graph adjacency matrix*

Description

Construct a graph adjacency matrix

Usage

```
make_graph(
  p = 70,
  type = c("proximity", "ring", "chain", "community", "disconnected")
)
```

Arguments

p	Number of nodes.
type	One of "proximity", "ring", "chain", "community", or "disconnected".

Value

A symmetric p x p binary adjacency matrix.

`run_one_replicate` *Run a single simulation replicate*

Description

Run a single simulation replicate

Usage

```
run_one_replicate(
  n = 100,
  p = 70,
  q = 500,
  graph_type = "proximity",
  signal_pattern = "smooth",
  fit_graph_type = NULL,
  graph_corruption = 0,
  s2 = 5,
  signal = 3,
  s_alpha = 10,
  alpha_strength = 5,
  smooth_noise = 0.2,
  threshold = 1e-04
)
```

Arguments

`n`	Sample size.
`p`	Number of exposures.
`q`	Number of instruments.
`graph_type`	Graph topology passed to `make_graph`.
`signal_pattern`	One of `"smooth"`, `"nonsmooth"`, or `"community"`.
`fit_graph_type`	Graph supplied to estimators. If NULL uses `graph_type`.
`graph_corruption`	Proportion of edges to corrupt. Default 0.
`s2`	Number of active nodes.
`signal`	Causal effect magnitude.
`s_alpha`	Number of invalid instruments.
`alpha_strength`	Invalid-IV direct-effect magnitude.
`smooth_noise`	Noise on the smooth signal pattern.
`threshold`	Coefficients below this are treated as zero.

Value

A data.frame with one row per method and columns Method, MSE, MCC, TPR, FPR, Selected.

`run_simulation` *Run a simulation study with multiple replicates*

Description

Run a simulation study with multiple replicates

Usage

```
run_simulation(
  B = 100,
  n = 100,
  p = 70,
  q = 500,
  graph_type = "proximity",
  signal_pattern = "smooth",
  fit_graph_type = NULL,
  graph_corruption = 0,
  s2 = 5,
  signal = 3,
  s_alpha = 10,
  alpha_strength = 5,
  smooth_noise = 0.2,
  threshold = 1e-04
)
```

Arguments

`B`	Number of Monte Carlo replicates.
`n`	Sample size.
`p`	Number of exposures.
`q`	Number of instruments.
`graph_type`	Graph topology passed to `make_graph`.
`signal_pattern`	One of `"smooth"`, `"nonsmooth"`, or `"community"`.
`fit_graph_type`	Graph supplied to estimators. If NULL uses `graph_type`.
`graph_corruption`	Proportion of edges to corrupt. Default 0.
`s2`	Number of active nodes.
`signal`	Causal effect magnitude.
`s_alpha`	Number of invalid instruments.
`alpha_strength`	Invalid-IV direct-effect magnitude.
`smooth_noise`	Noise on the smooth signal pattern.
`threshold`	Coefficients below this are treated as zero.

Value

A data.frame with 3*B rows, one per method per replicate.

Index