

Package ‘HVS’

August 19, 2026

Title Hypergraph Variable Selection

Version 1.0.1

Description

Performs hypergraph-based setwise variable selection with false discovery rate control (Organ, Kenney & Gu, 2026, <doi:10.48550/arXiv.2606.20514>). The idea is, in addition to selecting individual predictors when there is sufficient evidence, to also test all pairs of predictors, and when there is insufficient evidence to be sure which is the true predictor, it will select possibly overlapping pairs, for which there is strong evidence that at least one is a true predictor. The method is designed to control a generalised false discovery rate, where discoveries are counted based on the number of independent sets. The function of this package is similar to the 'hypergraph.sizing' package, but this package is optimised for faster computation in the case where we test all pairs of predictors. The package also includes functions for counting independent sets in a graph or hypergraph, either exactly or approximately. There is also a very limited function for isotonic regression, which is designed for fast computation in the specific case needed for hypergraph variable selection, rather than for general use.

Depends R (>= 3.2.3)

License GPL-3

Encoding UTF-8

NeedsCompilation yes

Maintainer Toby Kenney <tkenney@mathstat.dal.ca>

LinkingTo Rcpp

Author Toby Kenney [cre],
Sarah Organ [aut]

Repository CRAN

Date/Publication 2026-08-19 16:10:15 UTC

Contents

convex.hull	2
count.indep	3
count.indep.hyper	4

isotonic.regression	5
predict.isotonic	6
select.cutoff	7

Index	9
--------------	----------

convex.hull	<i>Convex hull of a sorted set of coordinates</i>
-------------	---

Description

Finds the upper convex hull of a sorted set of 2-dimensional coordinates.

Usage

```
convex.hull(x,y)
```

Arguments

x	A vector of x coordinates of points. This vector is assumed sorted.
y	A vector of the corresponding y coordinates. Assumed to be the same length as x.

Details

Finds the convex hull of the points $\{(x[1], -\text{Inf}), (x[1], y[1]), (x[2], y[2]), \dots, (x[n], y[n]), (x[n], -\text{Inf})\}$ where n is the length of the vector x . This function is designed to be used internally for isotonic regression, which is in turn used internally for the `select.cutoff` function. It is designed to be very efficient for this case, but not for general use - it is not flexible and has no error checking.

The method works by a divide-and-conquer approach. It divides the data in half, finds the convex hulls for both halves, then removes any middle points that are not in the combined convex hull. This results in a linear-time algorithm.

Value

A vector of indices, indicating which points are in the convex hull.

Author(s)

Toby Kenney

Examples

```
set.seed(1)
x<-seq_len(1000)
y<-runif(1000)*x*(1001-x)

hull<-convex.hull(x,y)
graphics::plot(x,y)
graphics::points(x[hull],y[hull],col="red",type='b')
```

count.indep

*Count the number of independent sets for a graph***Description**

Counts the number of independent sets (or equivalently vertex covers) in a graph.

Usage

```
count.indep(G,cutoff,method="exact",samp.size=1000)
count.vc(G,cutoff,method="exact",samp.size=1000)
```

Arguments

G	The graph, represented as a square matrix. The edges are the values smaller than the cutoff parameter.
cutoff	A value separating edges and non-edges of the graph.
method	A list of methods to use. The options are "exact" and "importance_sample". The "exact" method is computationally expensive, but counts the exact number of independent sets (or vertex covers). The "importance_sample" method uses a sequential importance sampling scheme to approximate the number of independent sets.
samp.size	For the approximate methods, this is the number of samples used to approximate the number of independent sets. Larger values take proportionately longer, but produce more accurate values.

Details

An independent set for a graph is a set that contains no edges. A vertex cover is a set that has non-empty intersection with every edge. The complement of an independent set is a vertex cover. These functions count the number of independent sets for a graph. The "exact" method exactly counts the number of independent sets by recursively partitioning into sets that contain a specific vertex and sets that do not. Since counting the number of independent sets in a graph is NP-hard, this is computationally expensive. The "importance_sample" method uses a random importance sampling scheme that sequentially samples independent sets starting from the empty set and adding one element each time. The samp.size parameter is the number of such sequences generated.

Both the exact and independent versions of this function are much faster than the corresponding hypergraph versions, due to a more efficient encoding of the graph structure.

Value

The number of independent sets (exact or approximate).

Author(s)

Toby Kenney, Sarah Organ

References

Hypergraph Variable Selection with False Discovery Rate Control

Sarah Organ, Toby Kenney, Hong Gu

<http://arxiv.org/abs/2606.20514>

Examples

```
set.seed(1)
G<-matrix(runif(400),20,20)
G<-pmin(G,t(G)) # make G symmetric

ind.set<-count.indep(G,0.05,method=c("exact","importance_sample"),)
```

count.indep.hyper	<i>Count the number of independent sets for a hypergraph</i>
-------------------	--

Description

Counts the number of independent sets (or equivalently vertex covers) in a hypergraph.

Usage

```
count.indep.hyper(H,method="exact",samp.size=1000)
count.vc.hyper(H,method="exact",samp.size=1000)
```

Arguments

H	The hypergraph, represented as a list with two components: vertices The number of vertices in the hypergraph. edges a list of edges. Each edge is a vector of vertex numbers.
method	A list of methods to use. The options are "exact" and "approx". The "exact" method is computationally expensive, but counts the exact number of independent sets (or vertex covers). The "approx" method uses a sequential importance sampling scheme to approximate the number of independent sets.
samp.size	For the "approx" method, this is the number of samples used to estimate the number of independent sets. Larger values take proportionately longer, but produce more accurate values.

Details

An independent set for a hypergraph is a set that contains no edges. A vertex cover is a set that has non-empty intersection with every edge. The complement of an independent set is a vertex cover, so there are the same number of independent sets as vertex covers. These functions count the number of independent sets for a hypergraph. The "exact" method exactly counts the number of independent sets by recursively partitioning into sets that contain a specific vertex and sets that do not. Since

counting the number of independent sets in a hypergraph is NP-hard, this is computationally expensive. The "approx" method uses a random importance sampling scheme that sequentially samples independent sets starting from the empty set and adding one element each time. The samp.size parameter is the number of such sequences generated.

If the hypergraph has no edges with more than two vertices, it is usually much faster to convert to a matrix representation and use the count.indep (or count.vc) function on this graph.

Value

The number of independent sets (exact or approximate).

Author(s)

Toby Kenney, Sarah Organ

References

Hypergraph Variable Selection with False Discovery Rate Control
 Sarah Organ, Toby Kenney, Hong Gu
<http://arxiv.org/abs/2606.20514>

Examples

```
set.seed(1)
edge.orders<-rpois(15,1.6)+1

H<-list("vertices"=10,lapply(edge.orders,function(s){sample(seq_len(10),s)}))

ind.set<-count.indep.hyper(H,method=c("exact","importance_sample"))
```

isotonic.regression	<i>Isotonic Regression</i>
---------------------	----------------------------

Description

Merges two sorted vectors, then fits the decreasing function that minimises squared error.

Usage

```
isotonic.regression(x1,y1,x2=NULL,y2=NULL)
```

Arguments

x1	A vector of predictor values. This vector is assumed sorted.
y1	A vector of the corresponding response values. Assumed to be the same length as x1.
x2	A second vector of predictor values. This vector is assumed sorted.
y2	A vector of the corresponding values. Assumed to be the same length as x2.

Details

This function is designed to be used internally by the function `select.cutoff`. It is provided as it may provide a useful efficient implementation of isotonic regression, but it is not designed for general use.

The function merges the sorted vectors `x1` and `x2` into a single vector, and merges `y1` and `y2` in the same way. It then fits the decreasing vector `z` that minimises the total squared error $\sum((z-y)^2)$ where `y` is the merged vector of `y1` and `y2`. By setting `x2=y2=NULL`, this function can perform isotonic regression on a single data set.

The solution to isotonic regression can be shown to be given by blockwise averages of the original data, where the blocks correspond to the convex hull of the cumulative sums.

Value

An object of class "isotonic" with the following components

x The sorted `x1` and `x2` values with `Inf` appended.

y The fitted isotonic regression values with 0 appended.

data The original `y1` and `y2` values, corresponding to increasing order of the combined `x1` and `x2` vectors.

Author(s)

Toby Kenney

Examples

```
set.seed(1)
x1<-seq_len(1000)
y1<-rnorm(1000)+3*exp(-x1/500)
x2<-seq_len(100)*2.1001
y2<-rnorm(100)+3*exp(-x2/500)

ir<-isotonic.regression(x1,y1,x2,y2)
graphics::plot(ir$x,ir$y)
```

predict.isotonic

Interpolation from Isotonic Regression

Description

Interpolates a prediction from an isotonic regression object

Usage

```
## S3 method for class 'isotonic'
predict(object,newdata,...)
```

Arguments

object	The isotonic regression object, consisting of a list with three components: x A sorted vector of cut-off values for the step function. y The corresponding values for the fitted response variable. data The original response values used to fit the data.
newdata	The values for which the function is to be calculated.
...	Any additional arguments are ignored.

Details

For each element of newdata, this function identifies the first element of object\$x that is larger, and looks up the corresponding element of object\$y.

Value

A vector of corresponding values of the step function.

Author(s)

Toby Kenney

Examples

```
set.seed(1)
x1<-seq_len(1000)
y1<-rnorm(1000)+3*exp(-x1/500)
x2<-seq_len(100)*2.001
y2<-rnorm(100)+3*exp(-x2/500)

ir<-isotonic.regression(x1,y1,x2,y2)
fitted<-predict(ir,newdata=seq_len(10000)/10)
```

select.cutoff	<i>Select the cut-off using HVS</i>
---------------	-------------------------------------

Description

Selects a p-value cut-off for hypergraph variable selection on a graph.

Usage

```
select.cutoff(G,batchsize=100,nbatch=100,level=0.05,threshold=NULL)
```

Arguments

G	The graph, represented as a square matrix. The values in the matrix represent the p-values of hypothesis tests corresponding to the edges of the graph (with diagonal elements corresponding to loops).
batchsize	The size of each batch of points.
nbatch	The number of batches
level	The desired FDR control level. If threshold is set to a numeric value, then this value is overridden.
threshold	Either a numeric value or null or "PRDS". If the value is null or "PRDS", then the value is recalculated to control the gFDR at the specified level, either under no dependence assumptions, or under the PRDS assumption.

Details

This selects the largest cut-off c such that the number of vertices minus the log to base 2 of the number of independent sets of the graph with edges having values below this cut-off, is more than or equal to c times the parameter "threshold".

Value

An object of class `c("HVS", "GLSUP")` which includes the following components:

level The specified gFDR control level.

pv The p-values in increasing order.

sizing The sizing function applied to the hyperedges with smallest p-values.

rejected Logical vector indicating which p-values are less than the cut-off.

cut.off.slope The specified or calculated threshold.

pv.cut.off The p-value cut-off.

sizing.function The estimated sizing function from isotonic regression. This can be studied to determine the uncertainty in counting the number of sets.

Author(s)

Toby Kenney, Sarah Organ

References

Hypergraph Variable Selection with False Discovery Rate Control
 Sarah Organ, Toby Kenney, Hong Gu
<http://arxiv.org/abs/2606.20514>

Examples

```
set.seed(1)
G<-matrix(runif(400)/10,20,20)
G<-pmin(G,t(G)) # make G symmetric

cut.off<-select.cutoff(G,threshold="PRDS")
```

Index

`convex.hull`, [2](#)
`count.indep`, [3](#)
`count.indep.hyper`, [4](#)
`count.vc(count.indep)`, [3](#)
`count.vc.hyper(count.indep.hyper)`, [4](#)

`isotonic.regression`, [5](#)

`predict.isotonic`, [6](#)

`select.cutoff`, [7](#)