Package ‘causaleffect’

February 23, 2016

Version 1.2.4

Date 2016-02-23

Title Deriving Expressions of Joint Interventional Distributions and Transport Formulas in Causal Models

Author Santtu Tikka

Maintainer Santtu Tikka <santtuth@gmail.com>

Imports ggm, igraph, XML

Suggests R.rsp

VignetteBuilder R.rsp

Description Functions for identification and transportation of causal effects. Provides a conditional causal effect identification algorithm (IDC) by Ilya Shpitser and Judea Pearl, and algorithms for z-identifiability, transportability, z-transportability and meta-transportability of causal effects by Elias Bareinboim and Judea Pearl. All of the previously mentioned algorithms are based on a causal effect identification algorithm by Jin Tian.

License GPL-2

NeedsCompilation no

Repository CRAN

Date/Publication 2016-02-23 09:40:27

R topics documented:

causaleffect-package 2
aux.effect 3
causal.effect 4
get.expression 6
meta.transport 7
parse.graphml 8
transport 9

Index **12**

causaleffect-package *Deriving Expressions of Joint Interventional Distributions and Transport Formulas in Causal Models*

Description

Do-calculus is concerned with estimating the interventional distribution of some action from the observed joint probability distribution of the variables in a given causal structure. All identifiable causal effects can be derived using the rules of do-calculus, but the rules themselves do not give any direct indication whether the effect in question is identifiable or not. Shpitser and Pearl (2006a) constructed an algorithm for identifying joint interventional distributions in causal models, which contain unobserved variables and induce directed acyclic graphs. A highly similar algorithm was constructed earlier by Tian (2002). The algorithm of Shpitser and Pearl (2006a) can be seen as a repeated application of the rules of do-calculus and known properties of probabilities, and it ultimately either derives an expression for the causal distribution or fails to identify the effect, in which case the effect is unidentifiable. Shpitser and Pearl (2006b) also presented a generalized algorithm for identification of conditional causal effects. causaleffect provides an implementation of this algorithm. In addition to ordinary identifiability, implementations of several other algorithms in causal inference are provided. These include algorithms for z-identifiability, transportability, z-transportability and meta-transportability of causal effects by Bareinboim and Pearl (2012, 2013a, 2013b, 2013c).

Graphs

Every causal model and selection diagram is depicted as an igraph graph with distinct attributes and special notation. Any bidirected edge corresponding to an unobserved variable must be denoted by using two unidirected edges with a `description` attribute of value `"U"`. Here is an example describing a simple causal model with only two vertices, X and Y, and a bidirected edge between them.

```
> g <- graph.formula(X -+ Y, Y -+ X)
> g <- set.edge.attribute(graph = g,
+ name = "description", index = 1:2, value = "U")
```

For selection diagrams, the vertices that correspond to selection variables must have a `description` attribute of value `"S"`. Here is an example of a simple selection diagram with a selection node S pointing to a non-selection variable Y. Because S precedes Y in the "-+" notation, S is given index 1 in the vertex sequence.

```
> d <- graph.formula(S -+ Y)
> d <- set.vertex.attribute(graph = d,
+ name = "description", index = 1, value = "S")
```

Author(s)

Santtu Tikka <santtuth@gmail.com>

References

Bareinboim, E., Pearl J. 2012 Causal Inference by Surrogate Experiments: z-identifiability. *Proceedings of the 28th Conference on Uncertainty in Artificial Intelligence*, 113–120.

Bareinboim, E., Pearl J. 2013a A General Algorithm for Deciding Transportability of Experimental Results. *Journal of Causal Inference*, **1**, 107–134.

Bareinboim, E., Pearl J. 2013b Meta-Transportability of Causal Effects: A Formal Approach. *Proceedings of the 16th International Conference on Artificial Intelligence and Statistics*, 135–143.

Bareinboim, E., Pearl J. 2013c Causal Transportability with Limited Experiments. *Proceedings of the 27th AAAI Conference on Artificial Intelligence*, 95 - 101.

Pearl J. 2009 *Causality: Models, Reasoning and Inference*, New York: Cambridge University Press.

Shpitser I., Pearl J. 2006a Identification of Joint Interventional Distributions in Recursive semi-Markovian Causal Models. *Proceedings of the 21st National Conference on Artificial Intelligence*, **2**, 1219–1226.

Shpitser I., Pearl J. 2006b Identification of Conditional Interventional Distributions. *Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence*, 427–444.

Tian J. 2002 Studies in Causal Reasoning and Learning. Phd thesis, Department of Computer Science, University of California, Los Angeles.

`aux.effect` *Identify a causal effect using surrogate experiments*

Description

This function returns an expression for the joint distribution of the set of variables (`y`) given the intervention on the set of variables (`x`) using auxiliary experiments on a set (`z`) if the effect is identifiable. Otherwise an error is thrown describing the graphical structure that witnesses unidentifiability.

Usage

```
aux.effect(y, x, z, G, expr = TRUE, simp = TRUE)
```

Arguments

`y`	A character vector of variables of interest given the intervention.
`x`	A character vector of the variables that are acted upon.
`z`	A character vector describing the additional set available for manipulation.
`G`	An `igraph` object describing the directed acyclic graph induced by the causal model that matches the internal syntax.
`expr`	A logical value. If `TRUE`, a string is returned describing the expression in LaTeX syntax. Else, a list structure is returned which can be manually parsed by the function `get.expression`
`simp`	A logical value. If `TRUE`, a simplification procedure is applied to the resulting probability object. d-separation and the rules of do-calculus are applied repeatedly to simplify the expression.

Value

A character string or an object of class `probability` that describes the interventional distribution.

Author(s)

Santtu Tikka

References

Bareinboim, E., Pearl J. 2012 Causal Inference by Surrogate Experiments: z-identifiability. *Proceedings of the 28th Conference on Uncertainty in Artificial Intelligence*, 113–120.

See Also

`parse.graphml`, `get.expression`

Examples

```
library(igraph)

# simplify = FALSE to allow multiple edges
f <- graph.formula(W -+ Z, Z -+ X, X -+ Y, W -+ Y, # Observed edges
  W -+ Y, Y -+ W, Z -+ Y, Y -+ Z, Z -+ X, X -+ Z, simplify = FALSE)

# Here the bidirected edges are set to be unobserved in graph g
# This is denoted by giving them a description attribute with the value "U"
# The first 4 edges correspond to the observed edges, the rest are unobserved

f <- set.edge.attribute(f, "description", 5:10, "U")
res <- aux.effect(y = "Y", x = "X", z = "Z", G = f)
```

`causal.effect` *Identify a causal effect*

Description

This function returns an expression for the joint distribution of the set of variables (`y`) given the intervention on the set of variables (`x`) conditional on (`z`) if the effect is identifiable. Otherwise an error is thrown describing the graphical structure that witnesses unidentifiability.

Usage

```
causal.effect(y, x, z = NULL, G, expr = TRUE, simp = TRUE)
```

Arguments

`y`	A character vector of variables of interest given the intervention.
`x`	A character vector of the variables that are acted upon.
`z`	A character vector of the conditioning variables.
`G`	An `igraph` object describing the directed acyclic graph induced by the causal model that matches the internal syntax.
`expr`	A logical value. If `TRUE`, a string is returned describing the expression in LaTeX syntax. Else, a list structure is returned which can be manually parsed by the function `get.expression`
`simp`	A logical value. If `TRUE`, a simplification procedure is applied to the resulting probability object. d-separation and the rules of do-calculus are applied repeatedly to simplify the expression.

Value

A character string or an object of class `probability` that describes the interventional distribution.

Author(s)

Santtu Tikka

References

Shpitser I., Pearl J. 2006 Identification of Joint Interventional Distributions in Recursive semi-Markovian Causal Models. *Proceedings of the 21st National Conference on Artificial Intelligence*, **2**, 1219–1226.

Shpitser I., Pearl J. 2006 Identification of Conditional Interventional Distributions. *Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence*, 427–444.

See Also

`parse.graphml`, `get.expression`

Examples

```
library(igraph)

# simplify = FALSE to allow multiple edges
g <- graph.formula(X -+ Y, Z -+ X, Z -+ Y , X -+ Z, Z -+ X, simplify = FALSE)

# Here the bidirected edge between X and Z is set to be unobserved in graph g
# This is denoted by giving them a description attribute with the value "U"
# The edges in question are the fourth and the fifth edge
g <- set.edge.attribute(graph = g, name = "description", index = c(4,5), value = "U")

res <- causal.effect(y = "Y", x = "X", G = g)
```

get.expression — *Get the expression of a probability object*

Description

This function converts an object of class `probability` returned by `aux.effect`, `causal.effect`, `meta.transport` or `transport` with `expr = FALSE` into a string which represents the probability distribution. Currently only LaTeX syntax is available.

Usage

```
get.expression(x)
```

Arguments

`x`	An object of class `probability` which is an internal list structure describing the interventional distribution.

Value

A character string that describes the resulting distribution in LaTeX syntax.

Author(s)

Santtu Tikka

See Also

`aux.effect`, `causal.effect`, `meta.transport`, `transport`

Examples

```
library(igraph)

# simplify = FALSE to allow multiple edges
g <- graph.formula(X -+ Y, Z -+ X, Z -+ Y , X -+ Z, Z -+ X, simplify = FALSE)

# Here the bidirected edge between X and Z is set to be unobserved in graph g
# This is denoted by giving them a description attribute with the value "U"
# The edges in question are the fourth and the fifth edge
g <- set.edge.attribute(graph = g, name = "description", index = c(4,5), value = "U")

x <- causal.effect(y = "Y", x = "X", z = NULL, G = g, expr = FALSE)
get.expression(x)
```

`meta.transport` *Derive a transport formula for a causal effect between a target domain and multiple source domains*

Description

This function returns an expression for the transport formula of a causal effect between a target domain and multiple source domains. The formula is returned for the interventional distribution of the set of variables (`y`) given the intervention on the set of variables (`x`). The multiple source domains are given as a list of selection diagrams (`D`). If the effect is non-transportable, an error is thrown describing the graphical structure that witnesses non-transportability. The vertices of any diagram in (`D`) that correspond to selection variables must have a description parameter of a single character "S" (shorthand for "selection").

Usage

```
meta.transport(y, x, D, expr = TRUE, simp = TRUE)
```

Arguments

`y`	A character vector of variables of interest given the intervention.
`x`	A character vector of the variables that are acted upon.
`D`	A list of `igraph` objects describing the selection diagrams in the internal syntax.
`expr`	A logical value. If `TRUE`, a string is returned describing the expression in LaTeX syntax. Else, a list structure is returned which can be manually parsed by the function `get.expression`
`simp`	A logical value. If `TRUE`, a simplification procedure is applied to the resulting probability object. d-separation and the rules of do-calculus are applied repeatedly to simplify the expression.

Value

A character string or an object of class `probability` that describes the transport formula.

Author(s)

Santtu Tikka

References

Bareinboim, E., Pearl J. 2013b Meta-Transportability of Causal Effects: A Formal Approach. *Proceedings of the 16th International Conference on Artificial Intelligence and Statistics*, 135–143.

See Also

`parse.graphml`, `get.expression`, `transport`

Examples

```
library(igraph)

# Selection diagram corresponding to the first source domain
# simplify = FALSE to allow multiple edges
d1 <- graph.formula(X -+ Z, W_1 -+ W_2, W_2 -+ Z,
  W_3 -+ Z, X -+ W_3, W_2 -+ X, Z -+ Y, # Observed edges
  S_1 -+ X, S_2 -+ W_2, S_3 -+ W_3, S_4 -+ Y, # Edges related to selection variables
  X -+ W_3, W_3 -+ X, X -+ W_2, W_2 -+ X, X -+ W_1,
  W_1 -+ X, W_1 -+ Z, Z -+ W_1, simplify = FALSE)

# Here the bidirected edges are set to be unobserved in the selection diagram d1
# This is denoted by giving them a description attribute with the value "U"
# The first 7 edges are observed and the next 4 are related to the selection variables
# The rest of the edges are unobserved
d1 <- set.edge.attribute(d1, "description", 12:19, "U")

# The variables "S_1", "S_2", "S_3" and "S_4" are selection variables
# This is denoted by giving them a description attribute with the value "S"
d1 <- set.vertex.attribute(d1, "description", 7:10, "S")

# Selection diagram corresponding to the second
# source domain is constructed in a similar fashion
d2 <- graph.formula(X -+ Z, W_1 -+ W_2, W_2 -+ Z, W_3 -+ Z,
 X -+ W_3, W_2 -+ X, Z -+ Y, # Observed edges
 S_1 -+ X, S_2 -+ W_2, S_3 -+ W_1,
 S_4 -+ Y, S_5 -+ Z, # Edges related to selection variables
 X -+ W_3, W_3 -+ X, X -+ W_2, W_2 -+ X, X -+ W_1,
 W_1 -+ X, W_1 -+ Z, Z -+ W_1, simplify = FALSE)
d2 <- set.edge.attribute(d2, "description", 13:20, "U")
d2 <- set.vertex.attribute(d2, "description", 7:11, "S")

# Combine the diagrams as a list
d.comb <- list(d1, d2)
res <- meta.transport(y = "Y", x = "X", D = d.comb)
```

`parse.graphml` *Prepare GraphML files for internal use*

Description

This function reads GraphML files created by a graphical editor, which describe directed acyclic graphs. The R package XML is utilized to parse the contents of the files to suit the internal format used by causal inference functions. Bidirected arcs are replaced by two unobserved unidirected arcs, and the resulting XML file is coerced into an igraph object. This function also serves as a wrapper for files that already correspond to the internal format. Names for the nodes of the graph can be supplied or read directly from the input file.

Usage

```
parse.graphml(file, format = c("standard", "internal"),
    nodes = c(), use.names = TRUE)
```

Arguments

`file`	The connection to read from.
`format`	A character constant describing how bidirected arcs are denoted in the GraphML file. Option `standard` corresponds to bidirected arcs that are notated with a graphical parameter describing an arrow at each end of the arc or no arrows at all. Option `internal` matches the format that `standard` graphs are coerced into. This option should be used only if all bidirected arcs in the graph are denoted by two unidirected arcs which have a description parameter of a single character "U" (shorthand for "unobserved").
`nodes`	A character vector that describes the names of the nodes in the graph. This is ignored if `use.names` is `TRUE`.
`use.names`	A logical value indicating whether the names of the nodes should be read from the file or not.

Value

An object of class igraph that describes the causal diagram. The parsed graph can now be used by `causal.effect`.

Author(s)

Santtu Tikka

`transport` *Derive a transport formula for a causal effect between two domains*

Description

This function returns an expression for the transport formula of a causal effect between two domains. The formula is returned for the interventional distribution of the set of variables (`y`) given the intervention on the set of variables (`x`) in a selection diagram (`D`). If the effect is non-transportable, an error is thrown describing the graphical structure that witnesses non-transportability. The vertices of (`D`) that correspond to selection variables must have a description parameter of a single character "S" (shorthand for "selection"). By default, every variable is available for intervention in the source. If only a subset of the variables is available, then the set (`z`) can be used to derive specific z-transportability.

Usage

```
transport(y, x, z = NULL, D, expr = TRUE, simp = TRUE)
```

Arguments

- `y` A character vector of variables of interest given the intervention.
- `x` A character vector of the variables that are acted upon.
- `z` A character vector of variables available for intervention. NULL value corresponds to ordinary transportability.
- `D` An `igraph` object describing a selection diagram in the internal syntax.
- `expr` A logical value. If `TRUE`, a string is returned describing the expression in LaTeX syntax. Else, a list structure is returned which can be manually parsed by the function `get.expression`
- `simp` A logical value. If `TRUE`, a simplification procedure is applied to the resulting probability object. d-separation and the rules of do-calculus are applied repeatedly to simplify the expression.

Value

A character string or an object of class `probability` that describes the transport formula.

Author(s)

Santtu Tikka

References

Bareinboim, E., Pearl J. 2013a A General Algorithm for Deciding Transportability of Experimental Results. *Journal of Causal Inference,*, **1**, 107–134.

Bareinboim, E., Pearl J. 2013c Causal Transportability with Limited Experiments. *Proceedings of the 27th AAAI Conference on Artificial Intelligence*, 95 - 101.

See Also

`parse.graphml`, `get.expression`, `meta.transport`

Examples

```
library(igraph)

# simplify = FALSE to allow multiple edges
d <- graph.formula(X -+ Z, Z -+ W, W -+ V, V -+ Y, S -+ V, # Observed edges
  X -+ Z, Z -+ X, V -+ Y, Y -+ V, X -+ Y, Y -+ X, simplify = FALSE)

# Here the bidirected edges are set to be unobserved in the selection diagram d
# This is denoted by giving them a description attribute with the value "U"
# The first five edges are observed, the rest are unobserved
d <- set.edge.attribute(d, "description", 6:11, "U")

# The variable "S" is a selection variable. This is denoted by giving it
# a description attribute with the value "S".
d <- set.vertex.attribute(d, "description", 6, "S")
```

```
res <- transport(y = "Y", x = "X", D = d)
```

Index

aux.effect, 3, *6*

causal.effect, 4, *6*
causaleffect *(*causaleffect-package*)*, 2
causaleffect-package, 2

get.expression, *4, 5*, 6, *7*, *10*

meta.transport, *6*, 7, *10*

parse.graphml, *4, 5*, *7*, 8, *10*

transport, *6, 7*, 9