# Package ‘cbl’

December 20, 2022

**Title** Causal Discovery under a Confounder Blanket

**Version** 0.1.3

**Description** Methods for learning causal relationships among a set of foreground variables X based on signals from a (potentially much larger) set of background variables Z, which are known non-descendants of X. The confounder blanket learner (CBL) uses sparse regression techniques to simultaneously perform many conditional independence tests, with complementary pairs stability selection to guarantee finite sample error control. CBL is sound and complete with respect to a so-called ``lazy oracle'', and works with both linear and nonlinear systems. For details, see Watson & Silva (2022) <arXiv:2205.05715>.

**License** GPL (>= 3)

**URL** https://github.com/dswatson/cbl

**Imports** data.table, foreach, glmnet, lightgbm

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**NeedsCompilation** no

**Author** David Watson [aut, cre] (<https://orcid.org/0000-0001-9632-2159>)

**Maintainer** David Watson <david.s.watson11@gmail.com>

**Depends** R (>= 3.5.0)

**Repository** CRAN

**Date/Publication** 2022-12-20 17:30:02 UTC

## R topics documented:

bipartite . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
cbl . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
epsilon_fn . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
l0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
minD . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6
r.TailProbs . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6
ss_fn . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7
sub_loop . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7

**Index** **8**

bipartite *Simulated data*

### Description

Simulated dataset of $n = 200$ samples with 2 foreground variables and 10 background variables. The design follows that of Watson & Silva (2022), with $Z$ drawn from a multivariate Gaussian distribution with a Toeplitz covariance matrix of autocorrelation $\rho = 0.25$. Expected sparsity is 0.5, signal-to-noise ratio is 2, and structural equations are linear. The ground truth for foreground variables is $X \rightarrow Y$.

### Usage

```
data(bipartite)
```

### Format

A list with two elements: `x` (foreground variables), and `z` (background variables).

### References

Watson, D.S. & Silva, R. (2022). Causal discovery under a confounder blanket. To appear in *Proceedings of the 38th Conference on Uncertainty in Artificial Intelligence*. *arXiv* preprint, 2205.05715.

### Examples

```
# Load data
data(bipartite)
x <- bipartite$x
z <- bipartite$z

# Set seed
set.seed(42)

# Run CBL
cbl(x, z)
```

`cbl` *Confounder blanket learner*

## Description

This function performs the confounder blanket learner (CBL) algorithm for causal discovery.

## Usage

```
cbl(
  x,
  z,
  s = "lasso",
  B = 50,
  gamma = 0.5,
  maxiter = NULL,
  params = NULL,
  parallel = FALSE,
  ...
)
```

## Arguments

| | |
|---|---|
| `x` | Matrix or data frame of foreground variables. |
| `z` | Matrix or data frame of background variables. |
| `s` | Feature selection method. Includes native support for sparse linear regression (`s = "lasso"`) and gradient boosting (`s = "boost"`). Alternatively, a user-supplied function mapping features `x` and outcome `y` to a bit vector indicating which features are selected. See Examples. |
| `B` | Number of complementary pairs to draw for stability selection. Following Shah & Samworth (2013), we recommend leaving this fixed at 50. |
| `gamma` | Omission threshold. If either of two foreground variables is omitted from the model for the other with frequency `gamma` or higher, we infer that they are causally disconnected. |
| `maxiter` | Maximum number of iterations to loop through if convergence is elusive. |
| `params` | Optional list to pass to `lgb.train` if `s = "boost"`. See `lightgbm::lgb.train`. |
| `parallel` | Compute stability selection subroutine in parallel? Must register backend beforehand, e.g. via `doMC`. |
| `...` | Extra parameters to be passed to the feature selection subroutine. |

## Details

The CBL algorithm (Watson & Silva, 2022) learns a partial order over foreground variables `x` via relations of minimal conditional (in)dependence with respect to a set of background variables `z`. The method is sound and complete with respect to a so-called "lazy oracle", who only answers independence queries about variable pairs conditioned on the intersection of their respective non-descendants.

For computational tractability, CBL performs conditional independence tests via supervised learning with feature selection. The current implementation includes support for sparse linear models (`s = "lasso"`) and gradient boosting machines (`s = "boost"`). For statistical inference, CBL uses complementary pairs stability selection (Shah & Samworth, 2013), which bounds the probability of errors of commission.

## Value

A square, lower triangular ancestrality matrix. Call this matrix `m`. If CBL infers that $X_i \prec X_j$, then `m[j, i] = 1`. If CBL infers that $X_i \preceq X_j$, then `m[j, i] = 0.5`. If CBL infers that $X_i \sim X_j$, then `m[j, i] = 0`. Otherwise, `m[j, i] = NA`.

## References

Watson, D.S. & Silva, R. (2022). Causal discovery under a confounder blanket. To appear in *Proceedings of the 38th Conference on Uncertainty in Artificial Intelligence*. *arXiv* preprint, 2205.05715.

Shah, R. & Samworth, R. (2013). Variable selection with error control: Another look at stability selection. *J. R. Statist. Soc. B*, *75*(1):55–80, 2013.

## Examples

```
# Load data
data(bipartite)
x <- bipartite$x
z <- bipartite$z

# Set seed
set.seed(123)

# Run CBL
cbl(x, z)

# With user-supplied feature selection subroutine
s_new <- function(x, y) {
  # Fit model, extract coefficients
  df <- data.frame(x, y)
  f_full <- lm(y ~ 0 + ., data = df)
  f_reduced <- step(f_full, trace = 0)
  keep <- names(coef(f_reduced))
  # Return bit vector
  out <- ifelse(colnames(x) %in% keep, 1, 0)
  return(out)
}

cbl(x, z, s = s_new)
```

`epsilon_fn` *Computer the consistency lower bound*

## Description

Computer the consistency lower bound

## Usage

```
epsilon_fn(df, B)
```

## Arguments

| | |
|---|---|
| `df` | Table of (de)activation rates. |
| `B` | Number of complementary pairs to draw for stability selection. |

`l0` *Feature selection subroutine*

## Description

This function fits a potentially sparse supervised learning model and returns a bit vector indicating which features were selected.

## Usage

```
l0(x, y, s, params, ...)
```

## Arguments

| | |
|---|---|
| `x` | Design matrix. |
| `y` | Outcome vector. |
| `s` | Regression method. Current options are `"lasso"` or `"boost"`. |
| `params` | Optional list of parameters to use when `s = "boost"`. |
| `...` | Extra parameters to be passed to the feature selection subroutine. |

`minD` *CPSS upper bound*

## Description

Compute the min-D factor of Shah & Samworth's Eq. 8 (2013). Code taken verbatim from Rajen Shah's personal website: http://www.statslab.cam.ac.uk/~rds37/papers/r_concave_tail.R.

## Usage

```
minD(theta, B, r = c(-1/2, -1/4))
```

## Arguments

| | |
|---|---|
| `theta` | Low rate threshold. |
| `B` | Number of complementary pairs for stability selection. |
| `r` | Of r-concavity fame. |

`r.TailProbs` *CPSS utility functions*

## Description

Compute the tail probability of an r-concave random variable. Code taken verbatim from Rajen Shah's personal website: http://www.statslab.cam.ac.uk/~rds37/papers/r_concave_tail.R.

## Usage

```
r.TailProbs(eta, B, r)
```

## Arguments

| | |
|---|---|
| `eta` | Upper bound on the expectation of the r-concave random variable. |
| `B` | Number of complementary pairs for stability selection. |
| `r` | Of r-concavity fame. |

`ss_fn` *Infer causal direction using stability selection*

## Description

Infer causal direction using stability selection

## Usage

```
ss_fn(df, epsilon, order, rule, B)
```

## Arguments

| | |
|---|---|
| `df` | Table of (de)activation rates. |
| `epsilon` | Consistency lower bound, as computed by `epsilon_fn`. |
| `order` | Causal order of interest, either `"ij"` or `"ji"`. |
| `rule` | Inference rule, either `"R1"` or `"R2"`. |
| `B` | Number of complementary pairs to draw for stability selection. |

`sub_loop` *Complementary pairs subsampling loop*

## Description

This function executes one loop of the model quartet for a given pair of foreground variables and records any disconnections and/or (de)activations.

## Usage

```
sub_loop(b, i, j, x, z_t, s, params, ...)
```

## Arguments

| | |
|---|---|
| `b` | Subsample index. |
| `i` | First foreground variable index. |
| `j` | Second foreground variable index. |
| `x` | Matrix of foreground variables. |
| `z_t` | Intersection of iteration-t known non-descendants for foreground variables `i` and `j`. |
| `s` | Regression method. Current options are `"lasso"` or `"boost"`. |
| `params` | Optional list of parameters to use when `s = "boost"`. |
| `...` | Extra parameters to be passed to the feature selection subroutine. |

# Index

∗ **datasets**
    bipartite, 2

bipartite, 2

cbl, 3

epsilon_fn, 5

l0, 5
lgb.train, *3*

minD, 6

r.TailProbs, 6

ss_fn, 7
sub_loop, 7