Package ‘r.blip’

October 14, 2022

Title Bayesian Network Learning Improved Project

Version 1.1


Depends R (>= 3.0.0)

Imports foreign, bnlearn (>= 4.0)

SystemRequirements Java (>= 1.5)

License LGPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

NeedsCompilation no

Author Mauro Scanagatta [aut, cre]

Maintainer Mauro Scanagatta <mauro@idsia.ch>

Repository CRAN

Date/Publication 2019-02-27 19:20:21 UTC

R topics documented:

blip .......................................................... 2
blip.learn .................................................. 2
**blip.learn**

---

**Description**

Used by most of the functions in the r.blip binding, provides access to the included jar file.

**Usage**

```r
blip(args)
```

**Arguments**

- `args` Vector of arguments to be passed to the jar

**Details**

The arguments vector is formatted in a system call to the included jar file. Should not be called directly by the user, unless you know exactly what you are doing. In that case, call directly the blip jar.

---

**blip.learn**

**Learns a BN**

---

**Description**

Fully learns a Bayesian networks.

**Usage**

```r
blip.learn(dat, scorer.method = "is", solver.method = "winasobs",
           indeg = 6, time = 3600, allocated = 80, scorefunction = "bic",
           alpha = 1, cores = 1, verbose = 0)
```
**Arguments**

- **dat**: dataframe from which to learn the parent sets (required)
- **scorer.method**: Method to be used for scoring the parent sets. Possible values: "is" (independence selection), "sq" (sequential selection). (default: is)
- **solver.method**: Method to be used for structure exploration. Possible values: "winasobs", "winobs", "asobs", "obs". (default: winasobs)
- **indeg**: Maximum number of parents (default: 6)
- **time**: Execution time (default: 3600)
- **allocated**: Percentage of the total execution time dedicated to parent set exploration (default: 80)
- **scorefunction**: Chosen score function. Possible choices: BIC, BDeu (default: bic)
- **alpha**: (if BDeu is chosen) equivalent sample size parameter (default: 1.0)
- **cores**: Number of machine cores to use. If 0, all are used. (default: 1)
- **verbose**: Verbose level (default: 0)

**Details**

The input data is required to be complete and discrete. Accordingly missing values in the input data.frame will be ignored, and all numeric values will be converted to integers.

**Value**

The learned Bayesian network in the bnlearn format.

**Examples**

```
bn <- blip.learn(child, time=3)
```

---

**blip.learn.tw**

*Learns a BN with a treewidth bound*

---

**Description**

Fully learns a Bayesian networks with a treewidth bound.

**Usage**

```
blip.learn.tw(dat, scorer.method = "is", solver.method = "kmax",
              treewidth = 5, time = 3600, allocated = 80,
              scorefunction = "bic", alpha = 1, cores = 1, verbose = 0)
```
### Arguments

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dat</td>
<td>dataframe from which to learn the parent sets (required)</td>
</tr>
<tr>
<td>scorer.method</td>
<td>Method to be used for scoring the parent sets. Possible values: &quot;is&quot; (independence selection), &quot;sq&quot; (sequential selection). (default: is)</td>
</tr>
<tr>
<td>solver.method</td>
<td>Method to be used for bounded-treewidth structure exploration. Possible values: &quot;kmax&quot;, &quot;kg&quot;, &quot;ka&quot;. (default: kmax)</td>
</tr>
<tr>
<td>treewidth</td>
<td>Maximum treewidth (default: 4)</td>
</tr>
<tr>
<td>time</td>
<td>Execution time (default: 3600)</td>
</tr>
<tr>
<td>allocated</td>
<td>Percentage of the total execution time dedicated to parent set exploration (default: 80)</td>
</tr>
<tr>
<td>scorefunction</td>
<td>Chosen score function. Possible choices: BIC, BDeu (default: bic)</td>
</tr>
<tr>
<td>alpha</td>
<td>(if BDeu is chosen) equivalent sample size parameter (default: 1.0)</td>
</tr>
<tr>
<td>cores</td>
<td>Number of machine cores to use. If 0, all are used. (default: 1)</td>
</tr>
<tr>
<td>verbose</td>
<td>Verbose level (default: 0)</td>
</tr>
</tbody>
</table>

### Details

The input data is required to be complete and discrete. Accordingly missing values in the input data.frame will be ignored, and all numeric values will be converted to integers.

### Value

The learned Bayesian network in the bnlearn format.

### Examples

```r
bn <- blip.learn.tw(child, treewidth=4, time=3)
```

---

**blip.scorer**  
*Parent set exploration*

### Description

Generates the cache of parent sets from a given data source.

### Usage

```r
blip.scorer(dat, method = "is", indeg = 6, time = 3600, scorefunction = "bic", alpha = 1, cores = 1, verbose = 0)
```
**Arguments**

dat dataframe from which to learn the parent sets. (required)
method Method to be used. Possible values: "is" (independence selection), "sq" (sequential selection). (default: is)
indeg Maximum number of parents (default: 6)
time Maximum Execution time (default: 3600)
scorefunction Chosen score function. Possible choices: BIC, BDeu (default: bic)
alpha (if BDeu is chosen) equivalent sample size parameter (default: 1.0)
cores Number of machine cores to use. If 0, all are used. (default: 1)
verbose Verbose level (default: 0)

**Details**

Usually the first step in the learning of a Bayesian network.
The input data is required to be complete and discrete. Accordingly missing values in the input data.frame will be ignored, and all numeric values will be converted to integers.

**Value**

Cache of parent sets

**Examples**

jkl <- blip.scorer(child, time=3)

---

**blip.solver**

**Structure Optimization**

**Description**

Find an optimal structure from the cache of parent sets

**Usage**

blip.solver(jkl, method = "winasobs", time = 3600, cores = 1, verbose = 0)

**Arguments**

jkl cache of pre-computed parent sets. (required)
method Method to be used. Possible values: "winasobs", "winobs", "asobs", "obs".
(time default: winasobs)
time Maximum Execution time (default: 3600)
cores Number of machine cores to use. If 0, all are used. (default: 1)
verbose Verbose level (default: 0)
Details
The input data is required to be complete and discrete. Accordingly missing values in the input
data.frame will be ignored, and all numeric values will be converted to integers.

Value
Structure

Examples
bn <- blip.solver(child.jkl, time=3)
child

---

**Description**

Dataset generated from the famous "child" network. Provided as an example of input data for learning a Bayesian network.

**Usage**

```r
data("child")
```

**Format**

The format is: chr "child"

**Details**

Space separated, integer values for each variable.

**Source**

http://www.bnlearn.com/bnrepository/discrete-medium.html#child

**Examples**

```r
data(child)
```

---

**child.jkl**

*Parent set cache for the child dataset*

---

**Description**

Parent set cache, taken from the "child" dataset.
read.jkl  Jkl reader

Description
Read a Jkl file (parent sets cache)

Usage
read.jkl(path, names)

Arguments
path  Path of the file to load
names  List of variable names

Value
the cache of parent sets

read.str  Structure reader

Description
Reads a str file (BN structure)

Usage
read.str(path, names)

Arguments
path  Path of the file to load
names  List of variable names

Value
the BN structure
write.jkl

Jkl writer (with names)

Description

Write a Jkl file (parent sets cache)

Usage

write.jkl(path, jkl)

Arguments

path Path of the file to write
jkl parent sets cache to write
Index

* datasets
  child, 7

* parentset
  child.jkl, 7

blip, 2
blip.learn, 2
blip.learn.tw, 3
blip.scorer, 4
blip.solver, 5
blip.solver.tw, 6

child, 7
child.jkl, 7

read.jkl, 8
read.str, 8

write.jkl, 9