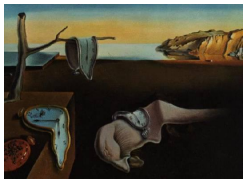

Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. *http://www.jstatsoft.org/*

Causal Inference using Graphical Models with the Package pcalg

Markus Kalisch ETH Zürich
Martin Mächler ETH Zürich
Diego Colombo ETH Zürich
Marloes H. Maathuis ETH Zürich
Peter Bühlmann ETH Zürich

Abstract

The abstract of the article.

Keywords: keywords, comma-separated, not capitalized, R.

1. Introduction

Understanding cause-effect relationships between variables is of primary interest in many fields of science. Usually, experimental intervention is used to find these relationships. In many settings, however, experiments are infeasible because of time, cost or ethical constraints.

Recently, we proposed and mathematically justified a statistical method (IDA) to obtain bounds on total causal effects based solely on observational data [ANNALS]. Futhermore, we recently presented an experimental validation of our method on a large-scale biological system [NATURE METHODS].

For further validation and broader use of this method, well documented and easy to use software is indispensible. Therefore, we wrote the R package **pcalg**, which incorporates the above mentioned method (IDA).

The objective of this paper is to introduce the R package **pcalg** and explain the main range of functions.

To get started quickly, we showi how two of the main functions can be used in a typical application. Suppose we have a system consisting of variables and many observations of this system. Furthermore, it seems plausible, that there are no hidden variables and no feedback loops in the underlying causal system. We are interested in the change of variable Y if we changed the variable X by intervention, i.e., we seek the causal effect of X on Y. To fix ideas, we have simulated an examle data set with $p = 8$ continuous variables with gaussian noise and $n = 5000$ observations, which we will now analyse. First, we load the data set.

```
> library(pcalg, lib.loc = "/u/kalisch/research/packages/pcalg.Rcheck")
```

```
> ## load data
> data(gaussianData)
```

In the next step, we use the function `pc` to produce an estimate of the underlying causal structure. Since this function is based on conditional independence tests, we need to define two things: First, a function that can compute conditional independence tests in a way that is suitable for the data at hand. For standard data types (gaussian, discrete and gaussian) we provide predefined functions which you can use. See the example section in the help file of `pc`. Secondly, we need a summary of the data (sufficient statistic) on which the conditional independence function can work. Each conditional independence test can be performed at a certain significance level `alpha`. This can be treated as a tuning parameter.

```
> ## use predefined test for conditional independence on gaussian data
> indepTest <- gaussCItest
> ## the functin gaussCItest needs as input the correlation matrix C and
> ## the sample size n
> suffStat <- list(C = cor(dat), n = 5000)
> ## estimate the causal structure
> pc.fit <- pc(suffStat, indepTest, p = 8, alpha = 0.01)

> ## plot the resulting causal structure
> plot(pc.fit)
```

As can be seen in the plot, there are directed and bidirected edges in the estimated causal structure. The directed edges show the presence and direction of direct causal effects. The direction of the bidirected edges, however, could not be decided by our method. Thus, they represent some uncertainty in the resulting model. A fundamental property of our method is, that some uncertainty of this kind sometimes remains, even if an infinite amount of data is available.

Based on the inferred causal structure, we can estimate the causal effect of an intervention. Suppose, we would increase variable 1 by one unit, how much would variable 6 increase? Since the causal structure was not identified perfectly, we cannot expect to get a unique number. Instead, we will get a set of possible causal effects. This set can be computed by using the function `ida`.

```
> ida(1, 6, cov(dat), pc.fit@graph)

[1] 0.3037948 0.2081007
```

Since we simulated the data, we know that the true value of the causal effect is 0.296. Thus, one of the two estimates is indeed close to the true value. Since both values are larger than zero, we can conclude, that variable 1 has a positive causal effect on variable 6 (note that we have no p-value to control the sampling error).

If we would like to know the effect of a unit increase in variable 1 on variables 3, 5 and 6, we could simply call `ida` three times. However, a faster way is to call the function `idaFast`, which was taylored for such situations. Each row shows the set of effects on the corresponding target variable.

```
> idaFast(1, c(3,5,6), cov(dat), pc.fit@graph)
```

```
    beta.tmp   beta.tmp
3 0.45481080 0.00000000
5 0.01623214 0.01927009
6 0.30379485 0.20810068
```

The true values for the causal effects are 0.441, 0, 0.296 for variables 3, 5 and 6, respectively. The first row of the output, corresponding to variable 3 is uninformative. Although one entry comes close to the true value, the other estimate is 0. Thus, we cannot be sure if there is a causal effect at all. The second row quite accurately indicates a causal effect that is very close to zero. The third row corresponds to variable 6 and was thus already discussed in the previous section on `ida`. !! Change package to give rownames !!

2. Methodological background

Our proposed method consists of two major steps. In the first step, the causal structure is estimated. This is done by estimating a graphical model. A graphical model is a map of the dependence structure of the data and can thus be an interesting object by itself. In the second step, we use the estimated causal structure and the do-calculus [PEARL] to calculate bounds on causal effects.

2.1. Estimating graphical models

Graphical models can be thought of as maps of dependence structures of a given probability distribution or a sample thereof (see for example **?**). In order to illustrate the analogy, let us consider a road map. In order to be able to use a road map, one needs two given factors. Firstly, one needs the physical map with symbols such as dots and lines. Secondly, one needs a rule for interpreting the symbols. For instance, a railroad map and a map for electric circuits might look very much alike, but their interpretation differs a lot. In the same sense, a graphical model is a map. Firstly, a graphical model consists of a graph with dots, lines and potentially arrowheads. Secondly, a graphical model always comes with a rule for interpreting this graph. In general, nodes in the graph represent (random) variables and edges represent some kind of dependence.

Without hidden and selection variables

An example of a graphical model is the Directed Acyclic Graph (DAG) model. The physical map here is a graph consisting of nodes and arrows (only one arrowhead per line) connecting the nodes. As a further restriction, the arrows must be directed in a way, so that it is not possible to trace a circle when following the arrowheads. The interpretation rule is d-separation, which is closely related to conditional independence. This rule is a bit more intricate and we refer the reader to **?** for more details.

- Definition of DAG model; ref. to Lauritzen
- Estimation methods: skeleton, pc; ref. to sgs, pc-paper

With hidden or selection variables

- AGs

- Estimation method: fci; ref. to sgs, zhang (completeness)

2.2. Estimating bounds on causal effects

Without hidden and selection variables

- if causal structure was known: do-calculus of pearl; result by Shpitser
- use GM as estimate for causal structure (causal markov property in pearl)
- ambiguity due to equivalence class -> get only set of possible effects
- local/global algorithm; equivalence; reference to Annals-paper
- example for application: see Nature Methods paper

With hidden and selection variables

estimating with latent variables present: not clear (?); perhaps cite zhang paper on theoretical results? Leave out?
ASSUMPTIONS?

3. Package pcalg

Two goals: Estimate graphical models and bounds on causal effects; link between the two; modularity in conditional independence tests; class as output with methods (own chapter?). In the following, we discuss the major functions of our package. Should I also mention that there are some deprecated functions?

3.1. skeleton

Explain options.

3.2. pc

Explain options.

3.3. ida

Explain options.

3.4. idaFast

Explain options.

3.5. fci

Explain options.

4. Examples

This is a Sweave example:

4.1. Estimating graphical models

Running example for gaussian data.

- simulate graph and data set
- use skeleton and pc
- summarize and plot resulting object; vary linewidth
- highlight modularity
- mention predefined functions for oracle, discrete data, binary data

4.2. Estimating bounds on causal effects

Continue running example starting from pc object of last section.

- use `ida` to estimate one effect with both local and global method
- use `idaFast` to compute effects on several x variables at the same time

5. Conclusion

Affiliation:

Markus Kalisch
Seminar für Statistik
ETH Zürich
8092 Zürich, Switzerland
E-mail: kalisch@stat.math.ethz.ch

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume VV, Issue II *Submitted:* yyyy-mm-dd
MMMMMM YYYY *Accepted:* yyyy-mm-dd