New permutation algorithms for causal discovery using ICA

Patrik O. Hoyer[1], Shohei Shimizu[1,2], Aapo Hyvärinen[1], Yutaka Kano[2], and Antti J. Kerminen[1]

[1] HIIT Basic Research Unit, Dept. of Comp. Science, University of Helsinki, Finland
[2] Graduate School of Engineering Science, Osaka University, Japan
http://www.cs.helsinki.fi/group/neuroinf/lingam/

Abstract. Causal discovery is the task of finding plausible causal relationships from statistical data [1, 2]. Such methods rely on various assumptions about the data generating process to identify it from uncontrolled observations. We have recently proposed a causal discovery method based on independent component analysis (ICA) called LiNGAM [3], showing how to completely identify the data generating process under the assumptions of linearity, non-gaussianity, and no hidden variables. In this paper, after briefly recapitulating this approach, we focus on the algorithmic problems encountered when the number of variables considered is large. Thus we extend the applicability of the method to data sets with tens of variables or more. Experiments confirm the performance of the proposed algorithms, implemented as part of the latest version of our freely available Matlab/Octave LiNGAM package.

1 Introduction

Several authors [1, 2] have recently formalized concepts related to causality using probability distributions defined on directed acyclic graphs. This line of research emphasizes the importance of understanding the process which generated the data, rather than only characterizing the joint distribution of the observed variables. The reasoning is that a causal understanding of the data is essential to be able to predict the consequences of interventions, such as setting a given variable to some specified value.

An interesting question within this theoretical framework is: 'Under what circumstances and in what way can one determine causal structure on the basis of observational data alone?'. In many cases it is impossible or too expensive to perform controlled experiments, and hence methods for discovering likely causal relations from uncontrolled data would be very valuable.

For continuous-valued data the main approach has been based on assumptions of linearity and gaussianity [1, 2]. Those assumptions generally lead only to a *set* of possible models equivalent in their conditional correlation structure. We have recently showed [3] that an assumption of *non-gaussianity* in fact allows the full model to be identified using a method based on independent component

analysis (ICA). However, this new method poses some challenging computational problems. In this paper we describe and solve these problems, allowing the application of the method to problems of high dimensionality.

The paper is structured as follows. In Section 2 we briefly describe the basics of LiNGAM, before focusing on the computational problems in Section 3. The proposed algorithms are empirically evaluated in Section 4. Conclusions are given in Section 5.

2 LiNGAM

Assume that we observe data generated from a process with the following properties:

1. The observed variables x_i, $i = \{1 \dots n\}$ can be arranged in a causal order $k(i)$, defined to be an ordering of the variables such that no later variable in the order participates in generating the value of any earlier variable. That is, the generating process is *recursive* [4], meaning it can be represented graphically by a *directed acyclic graph* (DAG) [2, 1].
2. The value assigned to each variable x_i is a *linear function* of the values already assigned to the earlier variables, plus a 'disturbance' (noise) term e_i, and plus an optional constant term c_i, that is
$$x_i = \sum_{k(j)<k(i)} b_{ij} x_j + e_i + c_i. \tag{1}$$
3. The disturbances e_i are all continuous random variables having *non-gaussian* distributions with non-zero variances, and the e_i are independent of each other, i.e. $p(e_1, \dots, e_n) = \prod_i p_i(e_i)$.

A model with these three properties we call a *Linear, Non-Gaussian, Acyclic Model*, abbreviated LiNGAM.

We assume that we observe a large number of data vectors $\mathbf{x}$ (containing the components x_i), and each is generated according to the above described process, with the same causal order $k(i)$, same coefficients b_{ij}, same constants c_i, and the disturbances e_i sampled independently from the same distributions. Note that the above assumptions imply that there are *no unobserved confounders* [2] (hidden variables). Spirtes et al. [1] call this the *causally sufficient* case.

To see how we can identify the parameters of the model from the set of data vectors $\mathbf{x}$, we start by subtracting out the mean of each variable x_i, leaving us with the following system of equations:

$$\mathbf{x} = \mathbf{B}\mathbf{x} + \mathbf{e}, \tag{2}$$

where $\mathbf{B}$ is a matrix that contains the coefficients b_{ij} and that could be permuted (by simultaneous equal row and column permutations) to strict lower triangularity if one knew a causal ordering $k(i)$ of the variables. (Strict lower triangularity is here defined as lower triangular with all zeros on the diagonal.) Solving for $\mathbf{x}$ one obtains

$$\mathbf{x} = \mathbf{A}\mathbf{e}, \tag{3}$$

where $\mathbf{A} = (\mathbf{I} - \mathbf{B})^{-1}$. Again, $\mathbf{A}$ could be permuted to lower triangularity (although not *strict* lower triangularity, actually in this case all diagonal elements will be *non-zero*) with an appropriate permutation $k(i)$. Taken together, equation (3) and the independence and non-gaussianity of the components of $\mathbf{e}$ define the standard linear independent component analysis (ICA) model [5, 6], which is known to be identifiable.

While ICA is essentially able to estimate $\mathbf{A}$ (and $\mathbf{W} = \mathbf{A}^{-1}$), there are two important indeterminacies that ICA cannot solve: First and foremost, the order of the independent components is in no way defined or fixed. Thus, we could reorder the independent components and, correspondingly, the columns of $\mathbf{A}$ (and rows of $\mathbf{W}$) and get an equivalent ICA model (the same probability density for the data). In most applications of ICA, this indeterminacy is of no significance and can be ignored, but in LiNGAM, we can and we have to find the correct permutation as described in Section 3 below.

The second indeterminancy of ICA concerns the scaling of the independent components. In ICA, this is usually handled by assuming all independent components to have unit variance, and scaling $\mathbf{W}$ and $\mathbf{A}$ appropriately. On the other hand, in LiNGAM (as in structural equation modeling, SEM [4]) we allow the disturbance variables to have arbitrary (non-zero) variances, but fix their weight (connection strength) to their corresponding observed variable to unity. This requires us to re-normalize the rows of $\mathbf{W}$ so that all the diagonal elements equal unity, before computing $\mathbf{B}$.

Our LiNGAM discovery algorithm [3] can thus be briefly summarized: First, use a standard ICA algorithm to obtain an estimate of the demixing matrix $\mathbf{W}$, permute its rows such that there are no zeros on its diagonal, rescale each row by dividing by the element on the diagonal, and finally compute $\mathbf{B} = \mathbf{I} - \mathbf{W}'$, where $\mathbf{W}'$ denotes the permuted and rescaled $\mathbf{W}$.

To find a causal order $k(i)$ we must subsequently find a second permutation, to be applied equally both to the rows and columns of $\mathbf{B}$, which yields strict lower triangularity.

3 Algorithms for solving the permutation problems

3.1 Permuting the rows of W

As pointed out above, because of the permutation indeterminancy of ICA, the rows of $\mathbf{W}$ will be in random order. This means that we do not yet have the correct correspondence between the disturbance variables e_i and the observed variables x_i. The former correspond to the rows of $\mathbf{W}$ while the latter correspond to the columns of $\mathbf{W}$. Thus, our first task is to permute the rows to obtain a correspondence between the rows and columns. If $\mathbf{W}$ were estimated exactly, there would exist one (and only one!) row permutation that would give a matrix with no zeros on the diagonal, and this permutation gives the correct correspondence [3]. Furthermore, finding the correct permutation would be trivial.

In practice, however, ICA algorithms applied on finite data sets will yield estimates which are only approximately zero for those elements which should be

exactly zero. Thus, we need to search for the correct permutation by minimizing a cost function which heavily penalizes small absolute values in the diagonal, such as $\sum_i 1/|\widetilde{W}_{ii}|$, where $\widetilde{\mathbf{W}}$ denotes the row-permuted $\mathbf{W}$.

An exhaustive search over all possible row-permutations is feasible only in relatively small dimensions [3]. For larger problems other optimization methods are needed. Fortunately, it turns out that the optimization problem can be written in the form of the classical *linear assignment problem*. To see this set $C_{ij} = 1/|W_{ij}|$, in which case the problem can be written as the minimization of

$$\sum_{i=1}^{n} C_{\phi(i),i} \tag{4}$$

where ϕ denotes the permutation to be optimized over. A great number of algorithms exist for this problem, with the best achieving worst-case complexity of $O(n^3)$ where n is the number of variables, see e.g. [7]. In our current implementation though, we simply use general-purpose linear programming software to find the optimum, which is good enough to solve problems involving tens of variables. Future implementations will use the more efficient algorithms.

3.2 Permuting B to get a causal order

Once we have obtained the correct correspondence between rows and columns of the ICA decomposition, calculating estimates of the b_{ij} is straightforward. First, we normalize the rows of the permuted matrix to yield $\mathbf{W}'$, and then calculate $\mathbf{B} = \mathbf{I} - \mathbf{W}'$ as described in Section 2 [3].

Although we now have initial estimates of all coefficients b_{ij} we do not yet have available a causal ordering $k(i)$ of the variables. Such an ordering (in general there may exist many if the generating network is not fully connected) is needed to achieve a directed acyclic graph, thus completing the estimation process. Essentially, after the ordering we can force half of the coefficients to equal zero such that the resulting network has no directed cycles.

A causal ordering can be found by permuting both rows and columns (using the same permutation) of the matrix $\mathbf{B}$ (containing the initial estimated connection strengths) to yield a strictly lower triangular matrix. If the estimates were exact, this would be a trivial task, using the following algorithm:

Algorithm A: Testing for DAGness, and returning a causal order if true

1. Initialize the permutation p to be an empty list
2. Repeat until $\mathbf{B}$ contains no more elements:
   (a) Find a row i of $\mathbf{B}$ containing all zeros, if not possible return **false**
   (b) Append i to the end of the list p
   (c) Remove the i:th row and the i:th column from $\mathbf{B}$
3. Return **true** and the found permutation p

However, since our estimates will not contain exact zeros, we will have to find a permutation such that setting the upper triangular elements to zero changes the matrix as little as possible. For instance, we could define our objective to be to minimize the sum of squares of elements on and above the diagonal, that is $\sum_{i \leq j} \widetilde{\mathbf{B}}_{ij}^2$ where $\widetilde{\mathbf{B}} = \mathbf{P}\mathbf{B}\mathbf{P}^T$ denotes the permuted $\mathbf{B}$, and $\mathbf{P}$ denotes the permutation matrix representing the sought permutation. In low dimensions, the optimal permutation can be found by exhaustive search. However, for larger problems this is obviously infeasible. Since we are not aware of any efficient method for exactly solving this combinatorial problem, we have taken another approach to handling the high-dimensional case.

Our approach is based on setting small (absolute) valued elements to zero, and testing whether the resulting matrix can be permuted to strict lower triangularity. Thus, the algorithm is:

Algorithm B: Finding a permutation of $\mathbf{B}$ by iterative pruning and testing

1. Set the $n(n+1)/2$ smallest (in absolute value) elements of $\mathbf{B}$ to zero
2. Repeat
   (a) Test if $\mathbf{B}$ can be permuted to strict lower triangularity (using Algorithm A above). If the answer is yes, stop and return the permuted $\mathbf{B}$
   (b) Additionally set the next smallest (in absolute value) element of $\mathbf{B}$ to zero

---

If in the problem, all the true zeros resulted in estimates smaller than all of the true non-zeros, this algorithm finds the optimal permutation. In general, however, the result is not optimal in terms of the above proposed objective; more elements are usually set to zero than would be needed. Fortunately, this is not a big problem because in sparse networks there are many more zeros in the coefficients than required by the acyclicity of the model, hence we would nevertheless like to prune out the small values from the estimated coefficients. See [3] for some discussion on pruning the estimated coefficients.

4 Experiments

In [3] we empirically verified the basic concept of LiNGAM by generating data from such models and estimating them using our method. However, because of the lack of efficient permutation algorithms we were limited to problems with small numbers of variables (8 variables or less). In the present paper we demonstrate that the method also works well in high dimensions by employing the permutation algorithms discussed in Section 3. All experimental code (including the precise code to produce Figure 1) is included in the LiNGAM code package, available at:

`http://www.cs.helsinki.fi/group/neuroinf/lingam/`

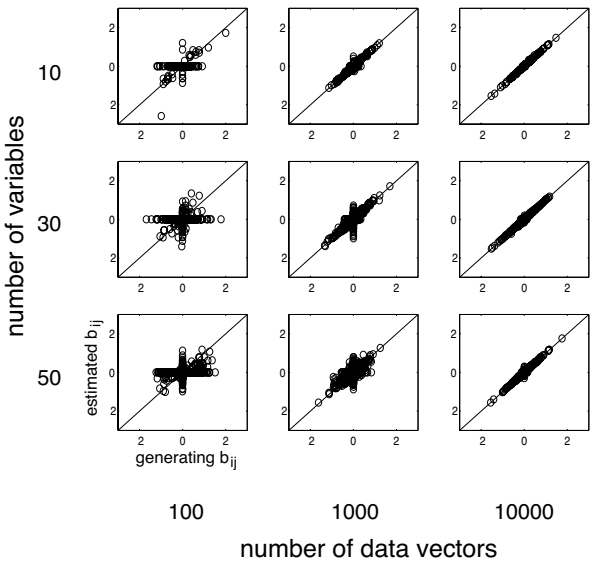

Fig. 1. Scatterplots of the estimated b_{ij} versus the original (generating) values. The different plots correspond to different numbers of variables and different numbers of data vectors. Although for small data sizes the estimation often fails, when there is sufficient data the estimation works essentially flawlessly, as evidenced by the grouping of the points along the diagonal.

We repeatedly performed the following experiment:

1. First, we randomly constructed a strictly lower-triangular matrix $\mathbf{B}$. Various dimensionalities (10, 30, and 50) were used. Both fully connected (no zeros in the strictly lower triangular part) and sparse networks (many zeros) were tested. We also randomly selected variances of the disturbance variables and values for the constants c_i.
2. Next, we generated data by independently drawing the disturbance variables e_i from gaussian distributions and subsequently passing them through a power non-linearity (raising the absolute value to an exponent in the interval [0.5, 0.8] or [1.2, 2.0], but keeping the original sign) to make them non-gaussian. Various data set sizes were tested. The e_i were then scaled to yield the desired variances, and the observed data $\mathbf{X}$ was generated according to the assumed recursive process (1).
3. Before feeding the data to the LiNGAM algorithm, we randomly permuted the rows of the data matrix $\mathbf{X}$ to hide the causal order with which the data was generated. At this point, we also permuted the generating coefficients, the c_i, as well as the variances of the disturbance variables to match the new order in the data.
4. Finally, we fed the data to our discovery algorithm, and compared the estimated parameters to the generating parameters. In particular, we made a

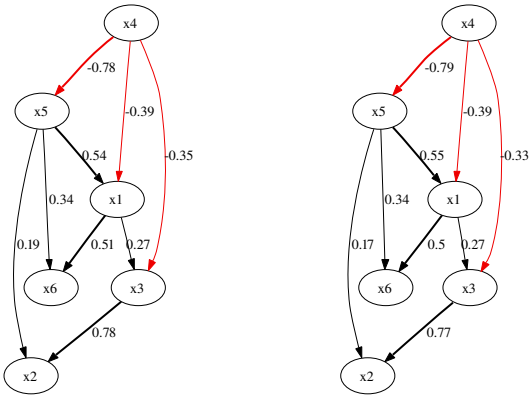

Fig. 2. Left: example original network. Right: estimated network. Graphs plotted using the latest version of the LiNGAM package which connects seamlessly to the free Graphviz software, a sophisticated tool for plotting graphs.

scatterplot of the entries in the estimated matrix **B** against the corresponding generating coefficients.

Since the number of different possible parameter configurations is limitless, we feel that the reader is best convinced by personally running the simulations using various settings. This can be easily done by anyone by downloading our software and running it using Matlab or the freely available Octave software. Nevertheless, we here show some representative results.

Figure 1 gives combined scatterplots of the elements of **B** versus the generating coefficients. The different plots correspond to different dimensionalities (numbers of variables) and different data sizes (numbers of data vectors), where each plot combines the data for a number of different network sparseness levels and non-linearities. Although for very small data sizes the estimation often fails, when the data size grows the estimation works practically flawlessly, as evidenced by the grouping of the datapoints onto the main diagonal.

In summary, the experiments verify that the new algorithms are able to find the appropriate permutations even in high dimensions, and demonstrate that reliable estimation is possible even when the number of variables is large. Comparing with the experiments in [3] we note that for larger dimensions we clearly need more data, but the amounts of data required are still reasonable.

5 Conclusions

Developing methods for causal inference from non-experimental data (data which does not come from controlled, randomized experiments) is a fundamental problem with a very large number of potential applications. Although one can never fully prove the validity of a causal model from observational data alone, such

methods are nevertheless crucial in cases where it is impossible or very costly to perform experiments.

The estimation of linear causal models can be based purely on the covariance structure of the data [4, 1, 2] but such methods cannot in most cases distinguish between multiple equally possible causal models that all imply the same conditional correlation structure. We have recently shown [3] that an assumption of non-gaussianity of the disturbance variables allows the full causal model to be identified, and provided an algorithm for this estimation. The method is essentially a post-processing method of ICA results.

In this paper we have shown how to solve one of the main remaining problems with our LiNGAM method, that of finding the appropriate permutations when the number of variables is large. The proposed algorithms have been implemented in our freely available software package, and tested in thorough experiments. The code package has also been extended to include graph plotting capability (in combination with Graphviz), as Figure 2 demonstrates.

How well real-world causal processes fit our assumptions, in particular that of linearity, will be crucial to the success or failure of applications of LiNGAM. We are currently involved in testing the method on real-world data and comparing its power and usefulness with other causal discovery methods, such as those based purely on conditional correlation structure. For the most recent developments, please see the project webpage.

Acknowledgements The authors would like to thank Aristides Gionis, Heikki Mannila, and Alex Pothen for discussions relating to algorithms for solving the permutation problems. P.O.H. was supported by the Academy of Finland project #204826. S.S. was supported by Grant-in-Aid for Scientific Research from the Ministry of Education, Culture and Sports, Japan. A.H. was supported by the Academy of Finland through an Academy Research Fellow Position and project #203344.

References

1. P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. MIT Press, 2000.
2. J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2000.
3. S. Shimizu, A. Hyvärinen, Y. Kano, and P. O. Hoyer. Discovery of non-gaussian linear causal models using ICA. In *Proc. 21st Conference on Uncertainty in Artificial Intelligence (UAI-2005)*, pages 526–533, Edinburgh, Scotland, 2005.
4. K. A. Bollen. *Structural Equations with Latent Variables*. John Wiley & Sons, 1989.
5. P. Comon. Independent component analysis – a new concept? *Signal Processing*, 36:287–314, 1994.
6. A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. Wiley Interscience, 2001.
7. R. E. Burkard and E. Cela. Linear assignment problems and extensions. In P. M. Pardalos and D.-Z. Du, editors, *Handbook of Combinatorial Optimization - Supplement Volume A*, pages 75–149. Kluwer, 1999.