Proceedings of the 3rd International Workshop on Distributed Statistical Computing (DSC 2003)
March 20–22, Vienna, Austria *ISSN 1609-395X*
Kurt Hornik, Friedrich Leisch & Achim Zeileis (eds.)
http://www.ci.tuwien.ac.at/Conferences/DSC-2003/

Evaluating the Effect of Perturbations in Reconstructing Network Topologies

Florian Markowetz **Rainer Spang**

Overview

Many different Bayesian network models have been suggested to reconstruct gene expression networks from microarray data. However, little attention has been payed to the effects of small sample size and the stability of the solution. We engage in a systematic investigation of these issues.

As a starting point for further research we introduce the κ-network. It is a small Bayesian network model (5 nodes with three states) in which a parameter κ controls the conditional probability distributions of the nodes. With data sampled from this model, we evaluate the effects of different sample sizes and of data being derived from active perturbations on the reconstruction of the origninal network topology.

1 Introduction

A genetic network is a set of genes in which individual genes influence the activity of other genes. The core task in identifying genetic networks is to distinguish direct from indirect regulatory interactions. Static and dynamic Bayesian network models have been suggested to reconstruct gene expression networks from microarray data Friedman et al. (2000); Murphy and Mian (1999); Yoo et al. (2002).

A Bayesian network is a graph-based representation of a joint probability distribution that captures properties of conditional independence between variables. This representation consists of two components. The first component is a directed acyclic graph (DAG), where the nodes represent genes and arrows between nodes indicate that one gene directly regulates the expression of another gene. The second component describes a conditional distribution for each node given its parents in the graph.

2 Learning network structure

The methods to build Bayesian networks from observational data can be divided into two classes: methods that use a scoring function to evaluate how well the network matches the data Friedman et al. (2000); Heckerman (1997), and methods that perform tests for conditional independence on the observations Pearl (2000); Spirtes et al. (2000).

The biological interpretation of the graphs produced by these methods is hindered by the fact that the representation of a joint distribution in a Bayesian network is not unique. Many different networks with ambiguous edges can represent the same joint distribution. Equivalent networks have the same skeleton, but edges not participating in v-structures may change their direction (see Figure 1 for an example). They indicate totally different gene regulation pathways but are statistically equivalent: Even with infinitely many data we can not decide between them.

Learning an equivalence class of networks is how far we get by depending only on passive observations. To further resolve the structure we need information about the effect of interventions. This determines the directions of the edges between the perturbed node and its neighbors Tian and Pearl (2001). For both approaches biological data is easily obtainable. Microarray experiments provide a snapshot of the activity of several thousand genes simultaneously. Gene perturbation as a method to identify regulation pathways has a long tradition in biology.

Before working on real gene expression data we start with a simulation where we know the true network topology and can adjust for different conditional probabilities in the nodes. We are interested in the following topics:

- Stability of solution: is the DAG with maximal Bayes score singled out sharply, or are there other DAGs with almost the same high score?
- How many data are needed to correctly identify the underlying structure?
- Robustness against changes in the conditional distribution of the nodes.

3 Methods and data

Software. We use the Bayes Net Toolbox for Matlab written by Kevin Murphy Murphy (2001) which is available at `http://www.ai.mit.edu/~murphyk`. Some of the functions were slightly changed and adapted. The Matlab scripts used for our experiments can be obtained from the first author.

The κ-network. As a starting point for further research we investigate a small network of five nodes with three possible values (downregulated=1, normal=2, upregulated=3). Its topology is the same as in the well-known sprinkler network (see Figure 1).

In the design of the conditional probabilities of the nodes we introduce a parameter κ which adjusts the distribution between a uniform distribution over the three states and one clearly preferred state. The distribution of each node is multinomial and

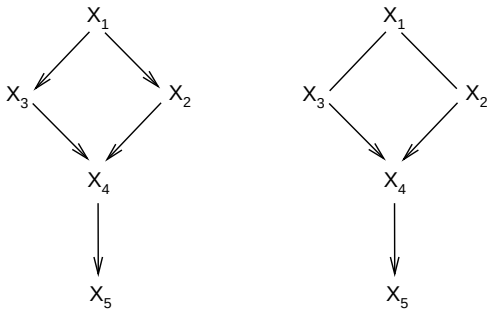

Figure 1: Left: topology of the sprinkler network. Right: the equivalence class of the sprinkler network described by a partially directed acyclic graph. It has the same skeleton and the same v-structure as the sprinkler network and encodes three different networks: the edges $X_2 \to X_4$, $X_3 \to X_4$ and $X_4 \to X_5$ are fixed by the v-structure $X_2 \to X_4 \leftarrow X_3$, but the edges $X_1 - X_2$ and $X_1 - X_3$ can point in both directions. Only $X_2 \to X_1 \leftarrow X_3$ is forbidden because this would create a new v-structure at X_1.

can be presented as a table with 3 columns and 3^{pa} rows, where pa is the number of parents. The columns correspond to the three possible values of a node X and the rows represent the possible configurations of parent nodes, e. g. for $pa = 2$ the nine rows stand for the configurations $(1,1), (1,2), \ldots, (3,3)$. The entry (i,j) is the probability of X being in state i given the j-th parent configuration.

We construct the distribution tables according to the rationale "κ·signal + $(1-\kappa)$·noise". For the "orphaned" node X_1 this results in the table $T_{(pa=0)}$ shown below. The noise term consists in a 1×3-matrix where all entries are equal to $\frac{1}{3}$.

$$T_{(pa=0)} = \kappa \cdot \begin{pmatrix} 0 & 1 & 0 \end{pmatrix} + \frac{1-\kappa}{3} \cdot \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}$$

The nodes X_2, X_3 and X_5 have a single parent node. The signal matrix propagates the parental state. The noise term is a 3×3-matrix abbreviated by $\frac{1}{3}(\text{ ones })_{3\times 3}$.

$$T_{(pa=1)} = \kappa \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} + \frac{1-\kappa}{3} \cdot (\text{ ones })_{3\times 3}$$

Node X_4 has two parents. If both parents agree in their state, its value is propagated in the signal matrix; if they do not agree, the signal is split equally over the two

parental states.

$$T_{(pa=2)} = \kappa \cdot \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 1 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 1 \end{pmatrix} + \frac{1-\kappa}{3} \cdot (\text{ ones })_{9\times 3}$$

By varying the parameter κ we can adjust the conditional distributions continuously between a uniform distribution over the states ($\kappa = 0$: pure noise) to a deterministic propagation of parental states ($\kappa = 1$: pure signal).

Network size. An obvious shortcoming of this model is the small number of nodes. Real-world gene regulation networks are of course much bigger. But on the other hand, the small number of nodes allows an exhaustive search through all possible DAGs and we can do without heuristic search methods. Scoring all possible DAGs provides the "gold standard" with which to compare other learning strategies. We investigate how our ability to recover the true structure changes with varying κ and how much information is gained by interventional data.

Changing the conditional distribution of the nodes. We vary κ over the interval $[0, .9]$ in steps of $1/10$. We omit the pathological setting $\kappa = 1$, because without any random effects all the nodes are in the same state and thus the learned graph will be completely connected. For each value of κ we sample from the corresponding network two different datasets of equal size. The first data are observations without intervention. The second data are gathered after perturbing the network at each node in turn. The dataset sizes are 100, 50 and 25 observations in the first dataset, which correspond to 20, 10 and 5 interventions per node in the second dataset.

Bayesian structure learning. From this data we infer a network structure by scoring all possible network topologies according to their posterior probability Heckerman (1997). The highest scoring DAG is chosen as the best representation of the relations between the data samples. Using interventional data only a single DAG achieves the maximal score. Without interventions we can only learn equivalence classes of DAGs. Thus, depending on dataset size more than one DAG can be found with maximal score.

Quality of learned networks. From the inferred topologies we calculate the relative frequencies of edges. This results in a 5×5-matrix L, where $0 \leq L_{ij} \leq 1$ is the relative frequency of the edge from X_i to X_j. This matrix is compared to the adjacency matrix A of the true topology by the formula:

$$d(A, L) \;=\; \sum_{ij} |A_{ij} - L_{ij}|$$

Repetitions. We repeat the process of data sampling and Bayesian learning 5 times for each dataset size and take the mean M_d of the five values of $d(A, L)$ as a final result.

4 Results

The score distribution. A number of 5 nodes is small enough to allow a scoring of all possible network structures. Thus, we can get an overview of the whole score distribution. Figure 2 shows the sorted scores for all possible 29281 DAGs with 5 nodes for $\kappa = .6$ and 5 interventions per node. For other values of κ and different numbers of observations the plot looks almost the same. What we learn from Figure 2 is that the maximal score is singled out sharply. We do not see a plateau of score values, where many DAGs achieve almost the same high score.

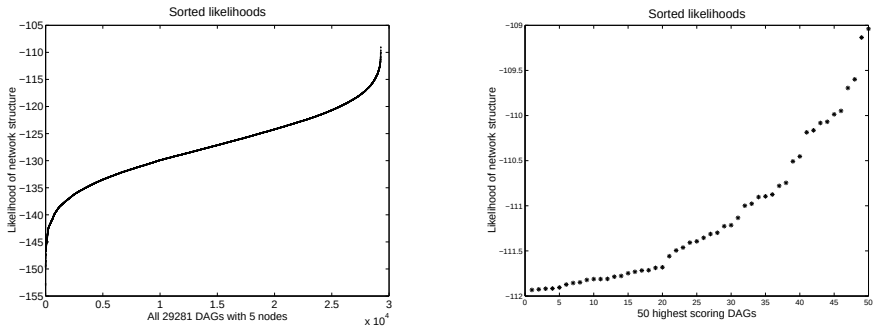

Figure 2: Left: The sorted likelihoods for all possible 29281 DAGs with 5 nodes ($\kappa = .6$, 5 interventions per node). Right: the 50 highest scoring DAGs - a zoom into the right end of the distribution.

For learning without interventions the general shape of the score distribution resembles Figure 2 with the difference, that the maximum value is achieved by more than one DAG. In this case, the actual number of DAGs depends on the dataset size. If a sufficient amount of data is supplied (several hundreds!), all three DAGs in the equivalence class of the sprinkler network are found (and no others). With less data, the number of maximal scoring DAGs varies in our experiments from two to five.

The effect of interventions. In Figure 3 the average number of false edges M_d is plotted for values of κ from 0 to .9 and datasets of size 25, 50 and 100. In each plot there are two lines: the dashed (red) line is the result of learning from passive observations only, while the solid (green) line results form learning with interventional data (5, 10, 20 interventions per node).

We see that for very noisy data learning from interventional data has no advantage

over learning from observations only (dashed and solid line agree very well for small values of κ). In the case of only 25 samples the two curves are almost the same over a wide range of κ-values. With increasing κ learning from interventional data becomes more efficient. With diminishing noise the interventions add more knowledge about the directions of the edges.

5 Discussion

We argue that incorporating interventional data not only leads to a higher number of correctly identified edges, but also reduces the required number of samples. But still big datasets are needed to learn the structure of even a small network like the κ-network.

From our experiments one message can be learned for the reconstruction of genetic networks: aim at small networks only and improve your accuracy by using data from gene perturbation experiments.

References

Nir Friedman, Michal Linial, Iftach Nachman, and Dana Pe'er. Using Bayesian networks to analyze expression data. *Journal of Computational Biology*, 7(3): 601–620, August 2000. URL http://citeseer.nj.nec.com/friedman99using.html.

David Heckerman. A Bayesian Approach to Causal Discovery. *Data Mining and Knowledge Discovery*, 1(1):79–119, 1997. URL http://citeseer.nj.nec.com/176059.html.

K. Murphy and S. Mian. Modelling gene expression data using dynamic Bayesian networks. Technical report, Computer Science Division, University of California, Berkeley, CA, 1999. URL http://citeseer.nj.nec.com/murphy99modelling.html.

Kevin Murphy. The Bayes Net Toolbox for Matlab. *Computing Science and Statistics*, 33, 2001. URL http://www.ai.mit.edu/~murphyk/Papers/bnt.ps.gz.

Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, Cambridge, 2000. URL http://bayes.cs.ucla.edu/BOOK-2K/index.html.

Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search*. MIT Press, Cambridge, MA, second edition, 2000.

Jin Tian and Judea Pearl. Causal discovery from changes. In *Proceedings of UAI 2001*, pages 512–521, 2001. URL http://citeseer.nj.nec.com/tian01causal.html. Part 1 of a two-part paper.

C. Yoo, V. Thorsson, and G.F. Cooper. Discovery of causal relationships in a generegulation pathway from a mixture of experimental and oberservational DNA microarray data. In *Proceedings of Pacific Symposium on Biocomputing 7:498-509*, 2002. URL http://citeseer.nj.nec.com/528477.html.

Affiliation

Florian Markowetz, Rainer Spang Max-Planck-Institute for Molecular Genetics
Computational Molecular Biology
Ihnestrasse 63-73
D-14195 Berlin
Germany
E-mail: florian.markowetz@molgen.mpg.de, rainer.spang@molgen.mpg.de

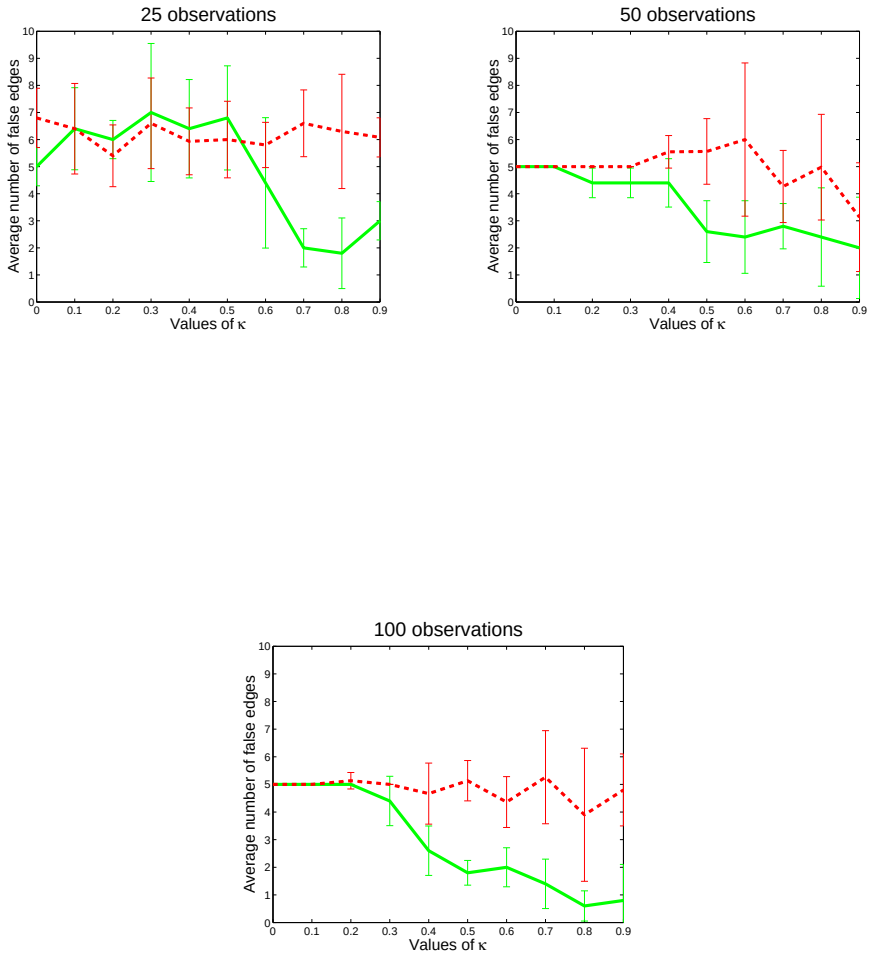

Figure 3: The average number of falsely predicted edges for increasing values of κ derived on datasets of size 25, 50 and 100. The dashed line is the result of learning from obervations only, while the solid line results form learning with interventional data. In all three plot, the dashed line stays constantly at a high error level of 5-6 false edges. For small values of κ the solid line is at the same level but drops with increasing κ. This is very clear in the lower plots, but even in the upper ones the solid line is never significantly higher than the dashed line.