ACTIVE LEARNING OF BAYESIAN NETWORK STRUCTURE IN A REALISTIC SETTING

Antti Larjo[1], *Harri Lähdesmäki*[1], *Marc Facciotti*[2], *Nitin Baliga*[2],
Olli Yli-Harja[1], *and Ilya Shmulevich*[2]

[1]Department of Signal Processing, Tampere University of Technology,
P.O. Box 553, FI-33101 Tampere, Finland
[2]Institute for Systems Biology,
1441 North 34th Street, Seattle, WA 98103, USA
antti.larjo@tut.fi, harri.lahdesmaki@tut.fi

ABSTRACT

Bayesian networks (BNs) are frequently used for modeling genetic regulatory networks. The structure of a static BN cannot in general be learnt unambiguously from observational data alone but interventions (i.e. knock-outs or over-expressions) are also required. These interventions can be difficult and costly to perform, thus calling for careful planning of experiments. Active learning methods can be used to suggest which interventions should be performed in order to increase our knowledge about the network structure maximally. Here, we utilize such a method for the first time in a realistic setting with measured wild-type and perturbed gene-expression and protein data and show the applicability and usability of the approach for designing biological experiments with maximal expected utility.

1. INTRODUCTION

Choosing which biological experiments to perform in order to benefit maximally from them is a highly non-trivial problem. The solutions to such a problem are context dependent: Trying to infer the dynamics of a system sets different demands on experimental design than when inferring the structure of a system, and will thus need to be addressed by different methods. Here we are interested in the problem of finding the structure of a biochemical subnetwork as efficiently as possible when the used model class is (causal) Bayesian networks. For demonstration, we consider learning both gene regulatory network and signaling network structures. We demonstrate the usability of a method to suggest maximally informative experiments.

2. METHODS

2.1. Bayesian networks

Given a set of random variables $\mathcal{X} = \{X_1, ..., X_n\}$, a Bayesian network is defined as a pair (G, θ), where G is a directed acyclic graph (DAG), which is a graphical representation of the conditional independencies between variables in $\mathcal{X}$, and θ is the set of parameters for the conditional probability distributions of these variables. The joint distribution over $\mathcal{X}$ factorizes according to G as

$$P\left(X_1, ..., X_n | G, \theta\right) = \prod_{i=1}^{n} P\left(X_i | Pa_G(X_i), \theta_i\right), \quad (1)$$

where $Pa_G(X_i)$ is the set of parents of node X_i in G, and θ_i the parameters for the distribution of X_i conditional on its parents.

In searching for the structure that most probably generated the data, of main interest is the posterior probability of a DAG given the data $P(G|D) = P(D|G)P(G)/P(D)$, where $P(G)$ is the prior probability of G,
$P(D) = \sum_{G'} P\left(D|G'\right) P(G')$ is the prior probability of data (sum goes over all possible DAG structures), and

$$P\left(D|G\right) = \int_{\theta} P\left(D|G, \theta\right) P\left(\theta|G\right) d\theta. \quad (2)$$

In this paper we only consider BNs having all the variables observed, discrete-valued and have multinomial conditional probability distributions (CPDs). We use uniform Dirichlet parameter priors since Dirichlet distribution is the conjugate prior of multinomials and makes it possible to obtain the closed form solution for Equation (2), which now becomes [1]

$$P(D|G) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(N'_{ij})}{\Gamma(N'_{ij} + N_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(N'_{ijk} + N_{ijk})}{\Gamma(N'_{ijk})}, \quad (3)$$

where N_{ijk} is the number of times the configuration $(X_i = k, Pa_G(X_i) = j)$ occurs in data D, N'_{ijk} are hyperparameters (a.k.a. pseudo-counts) of the Dirichlet distributions, $N_{ij} = \sum_k N_{ijk}$ and $N'_{ij} = \sum_k N'_{ijk}$, q_i is the number of different parent configurations, and r_i is the number of different states that node i can take.

Ideally, we would like to have the whole posterior distribution of DAGs and calculate our further analyses based on that (i.e. perform full Bayesian analysis). But since the number of different DAGs grows super-exponentially with n, evaluating the score (Equation (3)) for all possible structures is prohibitive for all but smallest of n ($n \leq 6$ or so). Instead, one is forced to resort to taking a sample of the posterior distribution with MCMC, as is done in this study.

An assumption we have to make is that the data is sampled from a probability distribution which can be represented with a Bayesian network (so called faithfulness assumption). In many real cases, this assumption is not likely to hold, especially for data from genetic networks (like in this study), where feedback loops are present. Still, we are obliged to make this approximation in order to be able to use a rigorous modeling approach.

2.2. Equivalence classes and interventional data

Given only observational (i.e. no interventions) data, it is generally impossible to learn the structure of a BN unambiguously because there is more than one structure producing the same combined probability distribution. Such sets of inseparable DAGs constitute equivalence classes, each of which consists of all the DAGs having the same v-structures[1] and otherwise the same structure when edge directions are ignored [2].

With interventions (i.e. forcing or ”clamping” a node or set of nodes to a certain value) we can break these classes, by inducing bias towards some of the alternatively possible structure(s). Forcing the value of a node determines the directions of edges adjacent to it and thus splits the equivalence classes into transition sequence (TS) equivalent structures [3]. With enough interventions, the size of the most probable TS equivalent class should reduce to one.

In gene networks, interventions can be either over-expressions, meaning that a gene is set to state ”on”, or knock-outs, corresponding to setting the gene ”off”. Since these interventions are based on biological mechanisms that are inherently stochastic, there is uncertainty in how well the intervention succeeds. However, here we take the interventions to be ideal.

2.3. Active learning

Active learning methods are designed to suggest which interventions should be made in order to maximally benefit from their effect of breaking equivalence classes or, more generally, to learn the structure of a BN with minimal cost of experiments.

Basically, two different approaches to selecting the perturbations have been presented: (i) those that break equivalence classes [4] and (ii) decision theoretic that aim to diminish our uncertainty (or increase information maximally) about some edges [5, 6]. These approaches are in fact closely related and complementary, since within an equivalence class the inability to say which direction an edge takes is, in other words, uncertainty about that edge.

We use the method presented by Murphy [5], where the expected utility of making an intervention a (which can be a plain observation, i.e. ”empty” intervention, or consist of setting the value of one or more nodes at a time) is defined as

$$V(a) = \sum_{G \in \mathcal{G}} \sum_{y \in \mathcal{Y}_{G,a}} P(y|G, a, D)P(G|D)U(G, a, y, D), \tag{4}$$

where $\mathcal{G}$ is our set of possible DAGs, $\mathcal{Y}_{G,a}$ denotes the set of possible observations that G can produce given that intervention a has been made. For the utility function $U(G, a, y, D)$ we use (assuming equivalent cost for each intervention) $\log P\left(G|a, y, D\right)$.

The best action is chosen from the set of possible actions $\mathcal{A}$ as the one with maximal utility
$a^* = \arg\max_{a \in \mathcal{A}} V(a)$. The optimal way of finding this action is by exhaustive enumeration.

Since the number of DAGs grows super-exponentially with the number of nodes, the exhaustive approach is practically unusable when $n > 6$. Therefore, stochastic sampling is used to obtain a sample from the posterior $P(G|D)$ which is then used in the above calculations.

Also, since the number of different observations a BN can produce is $\prod_{i=1}^{n} r_i$ (r_i is the number of discretization levels for node i), it quickly becomes too expensive to evaluate the above algorithm for all of them. Thus, we must again resort to sampling to keep the computing times reasonable. Sampling is done in this study in the same way as discussed in [5], by using importance sampling and drawing observations from a uniform distribution. The number of possible actions is rather small in our case so sampling is not needed for them.

3. RESULTS

3.1. Data

Our first dataset, which we refer to as the Halo dataset, consists of 242 gene expression measurements of 7 different transcription factors in *Halobacterium salinarum* [7, 8]. These transcription factors form the core of the transcriptional network in *H. salinarum* and are also believed to largely control the expression of each other, thus forming a small regulatory subnetwork. The dataset contains interventions (over-expressions) for all the 7 genes as well as normal observations (i.e. expression measurements without over-expressions). Therefore, this is an ideal dataset for our purposes.

The data was discretized into ternary values using a likelihood ratio statistic based model for detecting under- and over-expressed genes (with significance level 0.15) [9]. Some interventional measurements (8 in total) were removed due to having wrong discretization levels, implying most probably unsuccessful interventions.

The second dataset, which we call the Sachs dataset, consists of flow cytometry measurements from a signaling network with 11 nodes, of which 5 have been perturbed in some measurements [10]. These interventions contain both inhibitions and activations of the nodes, which should intuitively give the active learning a greater advantage over non-active learning than with the Halo dataset. The data

[1] a v-structure is a triplet (a, b, c) where $a \rightarrow b \leftarrow c$ and $a \nsim c$ (i.e. a and c are not joined).

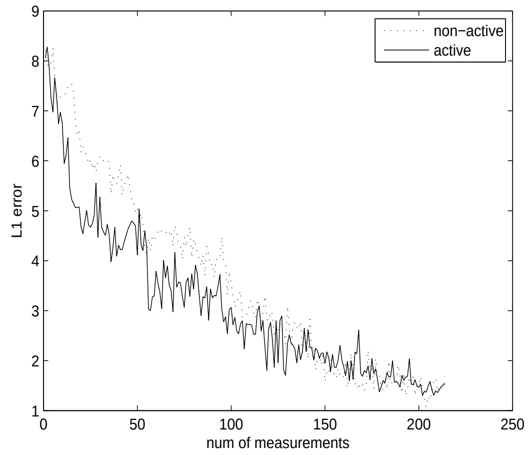

Figure 1. Using the Halo dataset, L_1 error was calculated for active and non-active learning methods by comparing to the structure derived from the ”true” posterior by taking edges with posterior probability > 0.5. Number of measurements are in addition to the initial 20 observations. Initial burn-in was $4 \cdot 10^5$, between-measurement burn-in was $2 \cdot 10^4$, graph sample size 10^4, and sampled observations 100. Results averaged from five different runs.

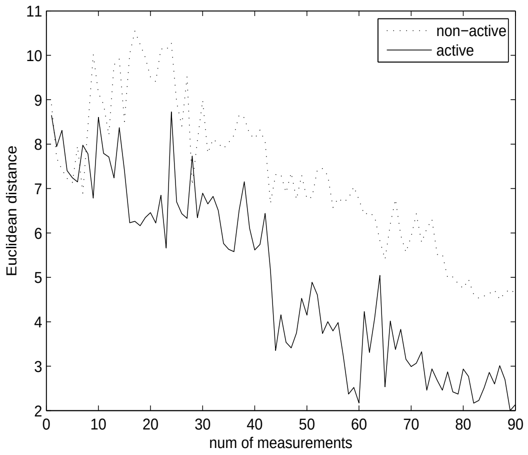

Figure 2. Euclidean distance between edge posterior probabilities calculated from the ”true” posterior distribution and either active or non-active learning methods when using the Sachs dataset. Number of measurements are in addition to the initial 40 observations. Initial burn-in was $2 \cdot 10^5$, between-measurement burn-in was 5000, graph sample size 5000, and sampled observations 300. Results averaged from four different runs.

was discretized into ternary values in the same way as in [10]. From the whole dataset we took a sample with 100 observational data points and 20 data points per intervention, totaling 220 measurements.

3.2. Active learning and random interventions

Instead of using the particle filter based updating done in [5], we used normal MCMC since it can be argued that it is what one would preferably use when there is plenty of computational time available between consequent measurements, as in, e.g., performing studies involving microarray measurements.

To compare the performances of the active and non-active learning methods, so called L_1 edge error was used [5, 6]

$$\begin{aligned} L_1(P_t) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} & I_{G^*}(X_i \to X_j)(1 - P_t(X_i \to X_j)) \\ & + I_{G^*}(X_i \leftarrow X_j)(1 - P_t(X_i \leftarrow X_j)) \qquad (5) \\ & + I_{G^*}(X_i \nsim X_j)(1 - P_t(X_i \nsim X_j)), \end{aligned}$$

where $P_t(\cdot) = P(\cdot|D_{1:t})$ is the posterior marginal probability of an edge given data points up to index t, and $I_{G^*}(c)$ is the indicatior function which takes value 1 if c is present in the true structure G^* and 0 otherwise. We also used the normal Euclidean distance between edge posterior probabilities as a measure of convergence towards the ”true” posterior distribution.

Each trial was initiated by taking a set of observations as initial data and, using this data, by running two MCMC chains in parallel for a long initial burn-in period. After this, samples were taken from both chains and the convergence of the chains was checked by comparing distributions of edge posterior probabilities calculated from both samples. When the distributions were similar, either sample was used as the initial sample for both active and non-active learners.

The active learning method proceeds by making at each step the measurement (intervention or observation) suggested by the active learning algorithm based on the sampled graphs, available measurements, sampled observations, and data collected so far. After each new measurement the chain is run for a between-measurement burn-in period and a new sample of graphs is taken. The non-active learning proceeds in the exact same way but instead of using an algorithm to suggest the next measurement, it just makes one randomly without replacement (i.e. takes one of the available measurements from the dataset).

To approximate the true posterior distributions, normal batch-style MCMC chains were run for the whole datasets and using very long burn-ins ($8 \cdot 10^5$ for the Sachs dataset and $2 \cdot 10^6$ for the Halo dataset) and big sample sizes ($2 \cdot 10^5$ for the Sachs dataset and $5 \cdot 10^5$ for the Halo dataset).

Figure 1 shows the results when using the Halo dataset. L_1 edge errors for both non-active and active learning methods were calculated by using as the reference structure the graph obtained by including only edges with posterior probability over 0.5 in the ”true” posterior distribution. Parameter values (sample sizes etc.) used are shown in the caption. Figure 2 shows the same using the Sachs dataset, except now Euclidean distances to the edge posterior probabilities of the ”true” posterior distribution were calculated for non-active and active learners. In this experiment we also took two similar measurements simultaneously instead of just one.

4. DISCUSSION

As can be deduced from Figures 1 and 2, the convergence towards the final results is faster with active learning than with non-active learning. Thus, using an active learning method to guide experimentation can result in savings in time and costs.

The performance of active learning methods has usually been assessed with simulated data. As shown here, the methods do not perform as convincingly with real data, due to possibly existing factors outside the targeted subsystem and the real systems containing cyclic regulatory relationships. Thus, it would be one step closer to reality if, e.g., simulated data from systems with hidden variables were used when comparing the methods.

Looking at the sequence of actions suggested by the active learning algorithm tells us what is probably intuitively clear: The most beneficial way is to mostly make interventional measurements rather than obtaining a lot of observational data. In the beginning of the investigations, however, it pays off to acquire (usually less costly) observations in order to get a solid basis for deciding which interventions to make. Even though part of the better performance of active learning over non-active can be explained by the fact that active learning suggests mostly interventions in the beginning while non-active learning samples uniformly from the set of interventional and observational measurements, the active learning should still (in the long run) overperform non-active due to choosing the order in which to make the interventions. This was also validated using simulated data (results not shown).

In order to be able to tell how many experiments to perform and when making more experiments produces no more benefit, a stopping criterion should be developed. A simple heuristic could be checking for the changes in posterior distribution between measurements and if there is no trend or bigger jumps in change, then it can be concluded at that point that more measurements tell us nothing new.

An alternative method of active learning by Pournara [4] approaches the problem by considering how to split the equivalence classes most efficiently. Although this is much faster than Murphy's method [5], the latter can perhaps be deemed to be more Bayesian, since it takes into account the distributions of generating observations. It is also not restricted to splitting equivalence classes but aims to minimize the conditional entropy of the posterior (or any other utility function). This reason, in particular, makes this method more general and precise by allowing it to, for example, suggest particular interventions several times if needed, instead of only suggesting the node with which to intervene without saying anything about how many measurements to take. However, because Murphy's method is computationally demanding and since sampling can affect the reliability/precision of the method, using the equivalence class based method becomes more attractive after about $n > 12$.

The active learning methods could also be developed towards better realistic applicability by making the cost of actions uneven and especially making the observations cheaper than interventions. The methods should also take into account the possibility of imperfect interventions. The idea of extending the methods to being able to suggest measurements from multiple different sources in an active learning fashion (for example by encoding them in priors) is also worth exploring.

5. ACKNOWLEDGMENTS

This work was supported by the Academy of Finland (application number 213462, Finnish Programme for Centres of Excellence in Research 2006-2011) and by grants R01 GM072855 and P50 GM076547 from NIH/NIGMS.

6. REFERENCES

[1] D. Heckerman, D. Geiger, and D. M. Chickering, "Learning Bayesian networks: The combination of knowledge and statistical data," *Mach. Learn.*, vol. 20, no. 3, pp. 197–243, 1995.

[2] T. Verma and J. Pearl, "Equivalence and synthesis of causal models," in *UAI '90: Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence*, New York, NY, USA, 1991, pp. 255–270, Elsevier Science Inc.

[3] J. Tian and J. Pearl, "Causal discovery from changes," in *UAI '01: Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*, San Francisco, CA, USA, 2001, pp. 512–521, Morgan Kaufmann Publishers Inc.

[4] I. Pournara and L. Wernisch, "Reconstruction of gene networks using Bayesian learning and manipulation experiments.," *Bioinformatics*, vol. 20, no. 17, pp. 2934–2942, Nov 2004.

[5] K. Murphy, "Active learning of causal Bayes net structure," Technical Report, University of California, Berkeley, USA, 2001.

[6] S. Tong and D. Koller, "Active learning for structure in Bayesian networks," in *Proceedings of the International Joint Conference on Artificial Intelligence*, 2001.

[7] R. Bonneau, M. T. Facciotti, D. J. Reiss, et al., "A predictive model for transcriptional control of physiology in a free living cell.," *Cell*, vol. 131, no. 7, pp. 1354–1365, Dec 2007.

[8] M. T. Facciotti, D. J. Reiss, M. Pan, et al., "General transcription factor specified global gene regulation in archaea," *Proceedings of the National Academy of Sciences*, vol. 104, no. 11, pp. 4630–4635, 2007.

[9] T. Ideker, V. Thorsson, A. F. Siegel, and L. E. Hood, "Testing for differentially-expressed genes by maximum-likelihood analysis of microarray data," *Journal of Computational Biology*, vol. 7, no. 6, pp. 805–817, 2000.

[10] K. Sachs, O. Perez, D. Pe'er, et al., "Causal protein-signaling networks derived from multiparameter single-cell data.," *Science*, vol. 308, no. 5721, pp. 523–529, Apr 2005.