Static Bayesian Modeling of Biological Time-Series Data

William H. Turkett, Jr.*

Department of Computer Science
Wake Forest University, Winston-Salem, NC 27109, USA

Abstract. Recent research into reconstructing biological networks has examined the use of dynamic Bayesian networks to model time-series data. While intuitively appealing, dynamic Bayesian network modeling makes assumptions about the properties of time-series data which may not hold for sparsely sampled datasets. This work argues that static Bayesian networks may be a more appropriate model for such datasets. Several exploratory results supporting this argument are presented on datasets previously cited in the dynamic Bayesian learning literature.

1 Introduction and Background

A key research problem in systems biology is the reconstruction of biological networks from time-series data. Reconstruction of gene transcription networks from microarray data and protein signaling networks from proteomic analyses are two motivating applications. A litany of computational approaches have been suggested in attacking the network reconstruction problem, with graphical models (Bayesian networks (BNs) in particular [2]) being a popular approach.

In using BNs for network reconstruction, one performs Bayesian network structure learning to find a model that fits the collected dataset. The learned structure, or a consensus model built over multiple learned structures, is hypothesized as the true network. This paper will focus on ab-initio search-based structure learning, where the space of BN models is searched to find the model(s) that best describes the collected biological data. Several recent works have explored learning from time-series data by using dynamic Bayesian networks (DBNs), models which incorporate specific temporal constraints. This work discusses assumptions inherent in the use of DBNs and suggests an alternative approach for small biological time-series datasets.

Dynamic Bayesian networks (DBNs) are a subset of general Bayesian networks which explicitly model the progression of a system over multiple timesteps. A DBN is specified by providing a graphical structure, indicating dependencies between variables in the domain, and probability tables, indicating prior probabilities or conditional probabilities for elements of the graph. Edges in the graph between timeslices model transitions between states after each timeslice event.

* This work was supported in part by the NSF-NIGMS Program in Mathematical Biology and NIH grant R01-GM075304.

Recent research [3][4] only allows edges across timesteps and transition models that are stationary and first order Markovian. The types of structures allowed by DBN modeling intuitively fit the types of network models one desires to reconstruct given time-series data. DBN learning is also relatively efficient, compared to standard BN learning, as it constrains the size of the network search space and does not require expensive sub-computations in the learning process (such as checking for cycles in the network). Various algorithms for learning DBN structures have been implemented - Yu et. al. [4] primarily use greedy search, while Husmeier has developed DBmcmc [3] to support MCMC-based search.

2 Dynamic and Static Modeling of Small Time-Series Datasets

The intent of this work is to inquire whether the DBN assumption of a stationary, first order Markov model is appropriate for learning from sparsely sampled datasets. Time series data is commonly reported for only a small number of time points (with a small number of technical and biological replicates performed). The sparse sampling of biological datasets raises concern about the suitability of DBNs. While the underlying system may be acting according to a first order Markov model, snapshot views of that system may not be.

To illustrate this issue, consider Genesim [4], a simulator that can be used to generate samples from a Markov process. Given a network structure, Genesim generates data according to the following equation:

$$Y_{t+1} = Y_t + A * (Y_t - T) + \epsilon \tag{1}$$

There are four factors in the Genesim model that contribute to the simulated measured expression of a gene. Y_t is used to model the current expression vector, with each gene's expression value bounded between zero and 100 and set randomly at the beginning of a simulation. Given a model with N genes, the matrix A is of size N^2 and holds inhibition and activation multiplicative factors between pairs of genes. The matrix T holds a threshold level (currently set at 50) used to influence the magnitude of the multiplicative effect (for example, an activator will more strongly influence a gene when the activator itself is at a high level). Finally, a small range of random noise is added to each output. See [4] for a more detailed description of the Genesim tool.

According to [4], near complete reconstruction of the network labeled **Network 1** in [4] is possible via DBN learning using 2000 data points. In that work, the data was collected by sampling one data point every 5 events during the evolution of the Genesim process. This network includes the interaction **11 inhibits 0 in the next timeslice**. Variable 11 is the only factor affecting 0 besides a small amount of random noise, so statistical analysis of the samples should show a fairly strong negative correlation between the expression vectors pairing variable 11 in time N and variable 0 in time N+1. The column labeled **Dynamic** in Table 1 indicates that this is the case, with a -0.84 Spearman correlation coefficient for the variable 11 and 0 vectors when 2000 data points are available.

Note, however, that when the system is randomly sampled over time, at 100, 50, and 25 points, the dynamic correlations are significantly reduced. Table 1 also provides the correlations for when a subset of data is collected at even intervals across the 2000 data points and when the first N samples are taken. Only when the first N samples are taken does a large negative correlation appear with the dynamic modeling approach.

Table 1. Spearman correlation coefficients for expression vectors

Number of Samples	Dynamic	Static
Full 2000	-0.84	-0.78
Random 100	-0.31	-0.77
Random 50	0.01	-0.83
Random 25	0.14	-0.71
Evenly Spaced 100	-0.07	-0.77
Evenly Spaced 50	0.236	-0.76
Evenly Spaced 25	0.094	-0.74
Initial N 100	-0.80	-0.81
Initial N 50	-0.44	-0.57
Initial N 25	-0.46	-0.68

This suggests that traditional static BN structure learning may be a more appropriate model given sparsely sampled data. Situations for which static modeling will be most effective are domains where effects exist long enough to be visible across multiple timesteps and where causal events are not one-shot triggers, but instead sustained causes. While a trigger would allow variable B to go to a particular state, given that A in some previous instance went to a particular state, a sustained cause would require that whenever variable A is in a particular state, variable B also exists in a corresponding state because of A. The **Static** column in Table 1 suggests that data generated by Genesim follows the sustained effects model described above. The Spearman correlation coefficient between the expression vectors for variables 11 and 0, with expression levels paired at the same timepoint instead of (N, N+1), remains high across both short and long sampling schemes.

Using static Bayesian network modeling implies the use of each time-step data point as a single sample from the system. Since this data is from a time course, it invalidates the static BN learning assumption that training samples are independent. However, due to the nature of the sampling process, it is believed that a violation of this assumption is less problematic than the first-order Markov assumption. While it is believed that network reconstruction will improve for some domains using static structure learning, various problems do exist with this alternative approach. Due to the acylic requirements of static BNs, feedback loops will be unable to be recovered. Additionally, due to the fact that Markov

equivalent networks are scored the same by common BN scoring methods (BIC and BDe), the edge directions in a network are not always recoverable using static learning.

3 Results

We have implemented MCMC-based BN and DBN structure learning algorithms in C++. Default parameters are BDe scoring, uniform structure priors, and max-in constraints of 3. An exploratory set of experiments have been performed to investigate static versus dynamic modeling. Two networks generated by the Genesim simulator, previously evaluated in [1] and [4], were initially examined. Each of these networks consisted of 20 variables, some of which performed a random walk while the rest were involved in a first order driven regulatory network. An additional network, the Zak network [3], was also examined, as it is believed to be biologically realistic. Arguably, this is a small set of initial experiments and a larger variety of network structures needs to be examined. These three networks were chosen as they represent networks which have previously been reported on in the dynamic Bayesian learning literature.

For both of the Genesim networks, 2000 datapoints were generated according to the sampling protocol specified in [4]. Subsets of 25, 50, and 100 samples were generated from this large set in three different ways: by random sampling without replacement, by taking evenly spaced samples, and by taking the initial N samples. These three approaches are an attempt to mimic potential wet-lab data collection schemes. MCMC search of the space of BNs consisted of 3 runs per dataset, with 200,000 burn-in (throw-away) and 200,000 collected samples. Edge occurence in 70% of the sampled networks was the threshold for whether or not an edge was included in the hypothesized model. There are 9 edges in the true Yu 1 network and 12 in the true Yu 2 network.

Tables 2 and 3 presents results for static and dynamic learning on the two Genesim datasets. The column **MCMC Edges** reports the total number of edges occuring greater than 70% of the time in the sampled networks. **True Positives**, in the second column, are counted without regard to directionality for static Bayesian learning. Static Bayesian learning techniques provide equivalent scores for Markov equivalent networks, leading to common sampling of networks in which the only change was for an edge to flip direction. Accordingly, in processing edge frequencies for static learning, counts of an edge in both directions are summed. This is not done for dynamic Bayesian learning, however, as dynamic learning explicitly assumes a forward-in-time direction. The lack of edge directions in the results from static learning indicate a need for further tests, such as perturbation experiments, to resolve edge directions. The column **FP Related** shows the number of false positives in a learned model that connect siblings or a grandparent-grandchild pair of the true network. These types of false positives occur with some frequency in Bayesian network learning - siblings more often in static Bayesian learning and grandparent-grandchild in DBN learning.

When the following parameters were used - initial N sampling, dynamic learning, and 50 or 100 samples - a large number of self links (a node connected to itself in the next timestep) are reported as high probability. The results for these parameters are marked with an asterisk in the result tables. While these links aren't explicitly stated as true edges in previous work [4], they are likely valid edges due to the nature of the generating process. These edges will not be picked up by any static Bayesian learning algorithms, as self links require cycles and cycles are illegal within a static Bayesian network. Because static Bayesian networks are required to be acyclical, static learning is not an appropriate approach for biological applications where the primary interest is to find feedback loops. If this is the intent of a lab gathering time course data, this work argues that such labs should collect significantly larger time courses and time courses that catch each process event, as these are the conditions for which dynamic learning works best in the experiments that were run.

Tables 2 and 3 support the argument that static learning is applicable when data is not able to be sampled at close, first order Markov intervals. There is a significant improvement in moving from dynamic to static learning for both randomly selected datapoints and evenly spaced, large gapped datapoints. Static learning even fares comparably when the initial N sampling method is used, the method that is closest to preserving the first order nature of the data. The other significant feature of the results is the large jump in total false positives in moving from 50 samples to 25 samples. This jump occurs in both dynamic and static learning. While a number of true positive edges are recovered using static learning, the false positive edges are not filtered out as desired, suggesting a threshold on learnability is being encountered.

The Zak dataset [3] consisted of 12 timepoint samples over 10 variables connected by 11 edges, all in a model that mimics several true structures seen in the biological literature. Husmeier [3], using DBN MCMC search, reports that four of the the top 11 edges learned are actual true positives. Using static Bayesian networks with 125,000 burn-in samples and 125,000 recorded samples, five true positives (absent direction), including the four found by Husmeier, were found.

References

1. Allen, E., Pecorella, A., Fetrow, J., John, D., and Turkett, W. Reconstructing networks using cotemporal functions. Proceedings of the 2006 ACM Southeast Conference (2006) 417–422
2. Friedman, N. Inferring cellular networks using probabilistic graphical models. Science **303** (2004) 799–805
3. Husmeier, D. Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks. Bioinformatics **19** (2003) 2271–2282
4. Yu, J., Smith, V., Wang, P., Hartemink, A., and Jarvis, E. Advances to Bayesian network inference for generating causal networks from observational biological data. Bioinformatics **20** (2004) 3594–3603

Table 2. True/false positive results from static and dynamic Bayesian learning on Yu et. al. Genesim network 1.

Sampling Style, Learning Style, Samples	MCMC Edges	True Positive	FP Related
Random,Dynamic,100	5	1	0
Random,Dynamic,50	1	0	0
Random,Dynamic,25	14	1	2
Random,Static,100	8	6	1
Random,Static,50	9	5	1
Random,Static,25	18	5	1
Evenly Spaced,Dyanamic,100	0	0	0
Evenly Spaced,Dynamic,50	5	0	0
Evenly Spaced,Dynamic,25	19	0	0
Evenly Spaced,Static,100	7	6	1
Evenly Spaced,Static,50	7	6	1
Evenly Spaced,Static,25	27	5	2
Initial N,Dynamic,100	28*	7	0
Initial N,Dynamic,50	24*	3	0
Initial N,Dynamic,25	18	1	0
Initial N,Static,100	23	4	1
Initial N,Static,50	13	2	0
Initial N,Static,25	17	1	0

Table 3. True/false positive results from static and dynamic Bayesian learning on Yu et. al. Genesim network 2.

Sampling Style, Learning Style, Samples	MCMC Edges	True Positive	FP Related
Random,Dynamic,100	9	4	2
Random,Dynamic,50	1	0	1
Random,Dynamic,25	19	1	0
Random,Static,100	10	5	4
Random,Static,50	9	4	4
Random,Static,25	16	3	3
Evenly Spaced,Dynamic,100	2	0	0
Evenly Spaced,Dynamic,50	3	0	0
Evenly Spaced,Dynamic,25	14	0	0
Evenly Spaced,Static,100	11	6	4
Evenly Spaced,Static,50	9	4	3
Evenly Spaced,Static,25	18	4	3
Initial N,Dynamic,100	24*	4	1
Initial N,Dynamic,50	20*	1	2
Initial N,Dynamic,25	12	0	0
Initial N,Static,100	19	5	3
Initial N,Static,50	18	1	2
Initial N,Static,25	18	2	0