ESANN'2006 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 26-28 April 2006, d-side publi., ISBN 2-930307-06-4.

Elucidating the structure of genetic regulatory networks: a study of a second order dynamical model on artificial data

Minh Quach[1] and Pierre Geurts[1,2] and Florence d'Alché-Buc[1] *

1- LAMI - CNRS UMR 8042 & Epigenomics project, Genopole
523, place des terrasses 91 EVRY, FRANCE

2- University of Liège - Dept of Electrical Engineering and Computer Science
Institut Montefiore, Sart Tilman, B28, 4000, Liège

Abstract. Learning regulatory networks from time-series of gene expression is a challenging task. We propose to use synthetic data to analyze the ability of a state-space model to retrieve the network structure while varying a number of relevant problem parameters. ROC curves together with new tools such as spectral clustering of local solutions found by EM are used to analyze these results and provide relevant insights.

1 Introduction

Learning of complex dynamical biological systems has emerged as a major challenge in post-genomics since the availability of DNA chips. We focus here on the learning of gene regulatory networks from time-series of gene expression [1, 2, 3]. Gene regulatory networks behave as a distributed and dynamical system whose variables are the genes that may regulate each other. Generative approaches such as dynamical bayesian networks have been developed in order to recover the structure of such networks. However while many works concern specific systems in model organisms such as *E. coli, S. cerevisiae*, it remains quite difficult to evaluate the ability of such methods to retrieve successfully the structure of a network. We thus need to study theoretically and empirically the capabilities and the limitations of our modeling methods. In this work, we use synthetic data to test one state-space model previously proposed in [4, 5]. We explore the ability of the model to retrieve the structure of a true network using ROC curves and area under the curve (AUC) while varying number of training time-series, length of sequences, noise, sampling-time, and presence of a hidden variable. In the special case of hidden variable, we found that the learning method is able to provide local minima solutions that can be gathered into two clusters using spectral clustering and graph kernel.

2 Learning structure with a state-space model

We study a model of genetic regulation, called Inertial Dynamic Bayesian Network (IDBN), previously introduced in [4, 5]. Due to space constraints, we only

*This research is funded by Genopole® (ATIGE). P.G. is a postdoctoral researcher at the CNRS and a scientific research worker at the FNRS.

give a short description of IDBN here and refer the reader to [4, 5] for details and a discussion of the biological relevance of this model.

Our state-space model represents gene expressions and their derivatives as an hidden process:

$$\left\{ \begin{array}{lcl} X_{t+1} & = & AX_t + \epsilon_{\mathbf{t}}^{\mathbf{h}} \\ Y_t & = & CX_t + \epsilon_{\mathbf{t}}^{\mathbf{o}} \end{array} \right. , \tag{1}$$

where X_t is the hidden state of the gene network at instant t corresponding to the n expression levels of the n genes and their n derivatives, and Y_t is the observed state of the network, composed of all observations of gene expression levels. $\epsilon_{\mathbf{t}}^{\mathbf{h}}$ and $\epsilon_{\mathbf{t}}^{\mathbf{o}}$ are isotropic Gaussian noise. Transition matrix A is defined as

$$A = \left[\begin{array}{cc} I & I \\ W - \Omega^2 & I - 2\Omega\Lambda \end{array} \right] \tag{2}$$

where I is the identity matrix of size $n \times n$, $W = (w_{ij})_{1 \leq i,j \leq n}$, $\Omega = \text{diag}(\omega_1, \ldots, \omega_n)$ and $\Lambda = \text{diag}\,(\lambda_1, \ldots, \lambda_n)$. C is the projection matrix $[I\ 0]$. We make the hypothesis that X_1 follows a Gaussian law of mean $\boldsymbol{\mu}_1$ and diagonal variance matrix $\sigma_1^2.I$. Parameters can be learned using EM and a MAP approach that allows to maximize log likelihood while minimizing the norm of w_{ij} parameters[1] [5].

Structure extraction In our approach we avoid the NP-hard problem of learning structure in Bayesian network because both structure and dynamics are encoded in the parameters. In some extent we can say that we delay the problem of structure identification to a post-processing of the obtained parameters. By doing this, we assume a strong interpretation of the transition matrix W that encode interactions.

Given a data set, several runs of EM from random initial conditions may reach different local solutions, leading to a distribution of the parameters w_{ij}. To extract a structure from these parameters, we make the hypothesis that gene j regulates gene i if w_{ij} is significantly different from zero according to this distribution. Assuming each w_{ij} follows a Gaussian distribution with mean $\bar{w}_{ij}$ and standard deviation s_{ij}, this gives the following z-test rules: (i) $\bar{w}_{ij} - \gamma * s_{ij} > 0$: gene j activates gene i (ii) $\bar{w}_{ij} + \gamma * s_{ij} < 0$: gene j inhibits gene i. γ is a parameter that reflects the confidence of the statistical test. For example, $\gamma = 1.96$ corresponds to a significance level of 0.05. Larger values of γ correspond to less connected networks.

In our experiments, we evaluate structures independently of γ by using ROC curves. A ROC curve is obtained by plotting the true positive rate versus the false positive rate for different values of γ. In our case, the true positive rate is defined as the proportion of true regulations that are correctly predicted with their sign[2]. The false positive rate is the proportion of non regulations that are wrongly predicted as regulations. A ROC curve is summarized by its area under the curve (AUC), with larger AUC meaning better performance.

[1] In all our experiments, the regularization parameter is kept fixed.

[2] Note that because the signs of regulations are taken into account, the true positive rate may never reach 1 for a given model even when all interactions are deemed significant ($\gamma = 0$).

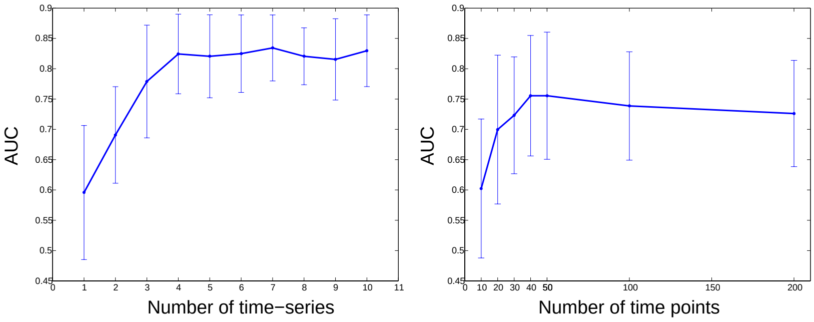

Fig. 1: AUC versus the number of time-series (left) and time points (right)

3 Empirical study of IDBN

Our experiments focus on the recovery of the network structure in various conditions. Another interesting criterion to evaluate a model is its ability to produce similar time series as the true model. Experiments not reported here show that our models are usually doing very well according to this criterion.

Generation of synthetic data. Our artificial regulatory networks are generated as scale-free networks that have been found to be relevant to gene networks. From the network structure, we then generate our parameter set as follows. The signs of non-zero w_{ij}'s are randomly chosen in $\{-1, +1\}$ and their magnitude are drawn from a uniform distribution $U(0, 0.04)$. Other parameters are randomly drawn from uniform distributions[3] until all eigenvalues of matrix A are inferior to 1 (to ensure the stability of the system). $\epsilon_{\mathbf{t}}^{\mathbf{h}}$ and $\epsilon_{\mathbf{t}}^{\mathbf{o}}$ follow a standard normal distribution. The means μ_1 of the initial state are sampled from $U(-100, 100)$ and σ_1 is set to 1. The model is then simulated from random initial states to collect S time-series of T time steps of the expressions corresponding to the N nodes. AUC values in Figures 1 and 2 are averaged over 10 random networks.

Number of time points and time-series. Because of high experimental costs, most time-series expression datasets contain only limited data, both in terms of the number of time points and in terms of the number of different time-series available for each gene. Hence, it is important that learning algorithms can work with very few data.

The left graph of Figure 1 shows the evolution of the AUC for networks of 10 nodes when we increase the number of time-series from 1 to 10 (with $T = 50$). Clearly, one time-series is not enough to obtain an interesting ROC curve. It does not put enough constraints on the learning algorithm and thus there are several solutions in terms of structure that perfectly fit the observed time series. On the other hand, 3 time-series already yield very good results with an AUC of 0.78. Increasing S above 5 does not bring further improvement with this respect.

[3] Exact parameter ranges are omitted for the sake of brevity.

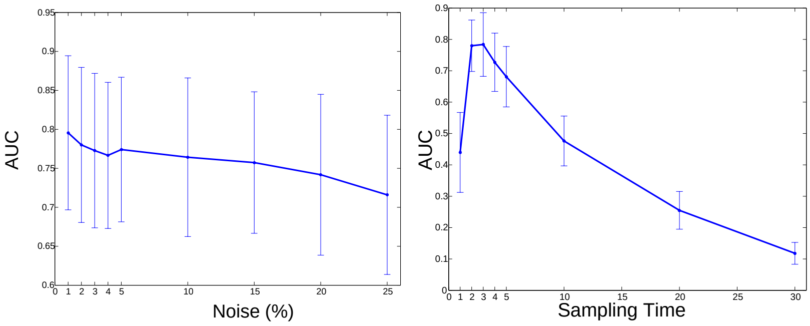

Fig. 2: AUC versus the noise level (left) and the sampling period (right)

The right graph of Figure 1 shows for $S = 3$ time-series the evolution of the AUC with T ranging from 10 to 200. The algorithm requires at least 20 time points to give good results. The AUC does not improve with more than 40 time points because our system is dampened and returns towards equilibrium.

Noise. Microarray expression data are usually very noisy. To study the effect of the noise on the performance of our algorithm, we increase the noise level on observed time-series from 1% to 25% of the average amplitude of the time series, using $N = 10$, $S = 3$, and $T = 50$. The left part of Figure 2 shows that performances decrease as the noise increases as expected. However, the decrease of AUC is quite slow.

Sampling period. Usually, the biologists choose the sampling time on the basis of their knowledge and the time scale of the phenomenon they want to study. Learning can only be possible if this sampling time is sufficiently small regarding the true process. In case of periodic signals, the Nyquist-Shannon theorem tells us that for a sampling period less than half the eigen period of the signal there is no loss of information. Although this theorem does not apply to our dampened oscillatory system, the effect of the sampling period can be observed with simulations.

To study this effect, we simulated a continuous version of the model with 10 nodes for a duration of 3000 minutes and we sampled the resulting continuous time series every 1 to 30 minutes over 50 time points. The right graph of Figure 2 shows that the AUC first increases and then decreases with the sampling rate. 50 time points with a rate of 1 does not capture enough information about the dynamics, which explains the low initial AUC. Good results are obtained with rates from 2 to 5 but the algorithm does not perform better than random at 10. Inspecting the time series, we observed that the minimal eigen period of the system precisely corresponds to a rate of 10. Sampling at this rate or higher leads to a loss of information that prevents reconstructing the dynamics.

Network size and structure. Error bars in Figures 1 and 2 shows that the performance of the algorithm depends heavily on the structure. This is confirmed by Figure 3 that plots the ROC curves (and their averages) corresponding to 10

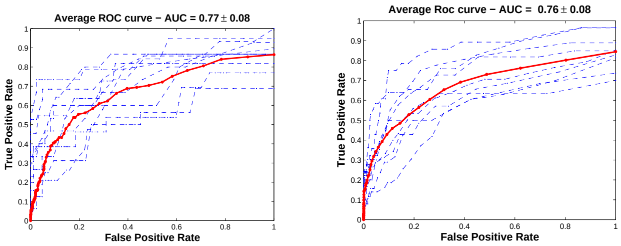

Fig. 3: ROC curves for 10 networks of 10 genes (left) and 20 genes (right)

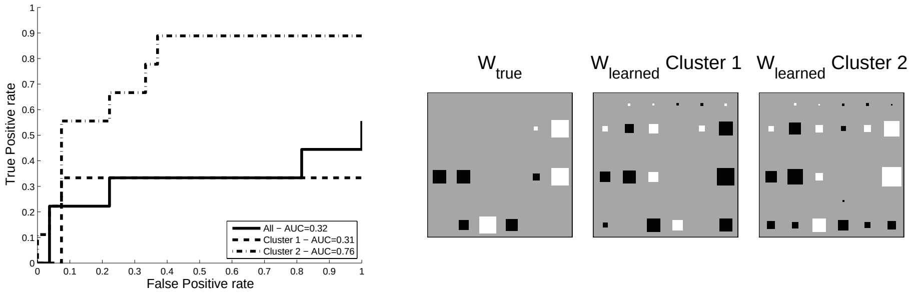

Fig. 4: Left, ROC curves obtained from all matrices and in the two clusters. Right, hinton graphs of the true weight matrix and the average matrices over the two clusters

random networks of 10 nodes (left) and 10 random networks of 20 nodes (right), with $T = 50$ and $S = 3$. However, average AUC are quite good in both cases. It is also interesting to note that doubling the number of nodes does not seem to affect average AUC and stability of the ROC curves.

4 Learning with hidden variables

One interesting feature of our model is that it can handle hidden variables, i.e. genes whose expressions are not observed but which influence the dynamics. However, in general, it is very difficult to retrieve the structure in the presence of hidden variables. Indeed, different behaviors of the hidden variables may all explain the behavior of the observed variables. In particular, for every solution, the solution obtained by reversing the weights implying a hidden variable is also acceptable. The EM algorithm can not choose between these equally likely alternatives and thus can get caught in any of the corresponding local optima depending of its initialization. This suggests that clustering local solutions may come up with a small set of plausible structures.

To study the ability of our algorithm to retrieve the structure in the presence of hidden variables, we simulated a random network of 6 genes with one hidden variable. The ROC curve obtained with $S = 3$ and $T = 50$ is given in the left graph of Figure 4. The result is very bad with an AUC of 0.32. We then apply clustering techniques on 100 solution matrices W obtained from different runs of the EM algorithm. We used kernel spectral clustering with an exponential graph kernel. The number of clusters was automatically determined using the ratio of between and within cluster distances. This procedure highlighted two very clear clusters of solutions. The ROC curves corresponding to the matrices in the two clusters are given in Figure 4. The first cluster obtains an AUC of 0.31 and the second one a much better AUC of 0.76. Figure 4 compares the weight matrices averaged over the two clusters (filtered with $\gamma = 0.5$) to the true weight matrix (left). The last column and the last row correspond to the hidden variable. Clearly, cluster 2 corresponds to the true solution while weight on the hidden variable are reversed in cluster 1.

So, although it is difficult to retrieve the structure in the presence of hidden variables, this example shows that clustering can be used to automatically determine a small set of possible solutions that can then be presented to the biologists for further analysis.

5 Conclusion and perspectives

The empirical study carried out in this paper has shown that the algorithm can provide useful information about the structure from very limited data (a few dozen time points and a couple of independent time series). The method is also robust to noise and sampling time to some extent. We have furthermore proposed a method to circumvent the problem of multiple solutions in the context of hidden variables based on clustering solution matrices.

While in this study the learned and true model belong to the same family, in the future, it would be interesting to apply our approach on artificial data generated from other gene regulation models to check its robustness to the model. We also intend to use non linear models of regulatory networks, closer to biochemical processes. In this case, the analysis of the Jacobian matrix will lead to the structure extraction. We expect that in this case also, the learning algorithm can come up with different families of solutions. Then clustering of solutions that was shown to be very relevant with hidden variables could be of use.

References

[1] P. D'haeseleer, S. Liang, and R. Somogyi. Genetic network inference: From co-expression clustering to reverse engineering. *BioInformatics*, 16(8):707–726, 2000.

[2] S. Kim, S. Imoto, and S. Miyano. Dynamic bayesian network and nonparametric regression for nonlinear modeling of gene networks from time series gene expression data. In *Proc. of CMSB 2003*, pages 104–113, 2003.

[3] A. Regev I. Nachman and N. Friedman. Inferring quantitative models of regulatory networks from expression data. *Bioinformatics*, Vol. 20 Suppl. 1:i248–i256, 2004.

[4] B-E. Perrin, L. Ralaivola, A. Mazurie, S. Bottani, J. Mallet, and F. D'Alché-Buc. Gene networks inference using dynamic bayesian networks. *Bioinformatics*, Vol. 19 Suppl. 2:i138–i148, 2003.

[5] F. d'Alché Buc, P.-J. Lahaye, B.-E. Perrin, L. Ralaivola, T. Vujasinovic, A. Mazurie, and S. Bottani. *Bioinformatics Using Computational Intelligence Paradigms*, chapter A dynamical system based on inertia principle for gene regulatory network modeling, pages 93–118. Springer, 2005.