Generative Learning of Dynamic Structures using Spanning Arborescence Sets

Anthony COUTANT and Céline ROUVEIROL
Laboratoire d'Informatique de Paris Nord (LIPN), UMR CNRS 7030, Université Paris 13
99 avenue Jean-Baptiste Clément, 93430 Villetaneuse, France

Corresponding author: firstname.lastname@lipn.univ-paris13.fr

Abstract *We are interested in the problem of generative learning of dynamic models from "fat" time series data (high #variables/#individuals ratio), leading to a high sensitivity of learned models to the dataset noise. We propose in this purpose a method computing a mixture of many highly biased but optimal spanning arborescences obtained from many perturbed versions of the original dataset, introducing variance to counterbalance the strong arborescence bias. The method is at the boundary between structure oriented Bayesian model averaging and recent work on density estimation using mixtures of poly-trees in a perturb and combine framework, transposed to a dynamic setting. In practice, preliminary results on the recent DREAM 8 challenge are promising.*

Keywords Network inference, Ensemble Learning, Model Averaging, Spanning Arborescences

1 Introduction

A major issue occuring in network inference algorithms from data is the one of *data fragmentation*, where the computation of frequencies is not reliable enough to avoid overfitting. This occurs more particularly when the dataset is scarce and suffers from a high variables/samples ratio, such as in many biological contexts.

Biological systems are also often characterized by their dynamic properties, and many biological phenomena under study are actually time evolving processes, e.g. the diauxic shift of the *Saccharomyces cerevisiae* yeast [1], for which available datasets can take the form of time series. This format often even reinforces the data fragmentation issue.

A possible strategy to prevent overfitting in such scarce situation is to reduce the learning algorithm variance, by reducing the number of possible models. This can be achieved for example by constraining the search space so that the resulting subspace have good properties, or by constraining the search algorithm so that a subset of possible models is reachable. Used alone, this strategy can find a good local, even global, optimum, but relatively to a potentally inadequate space where a good solution for the overall learning problem is unexistent. Another possibility is to find an asymptotic structure which is the result of a consensus between different models learned from the dataset [2,3]. This way, the lack of sufficient statistics on data is partly offset by an attempt to compute sufficient statistics on potentially very noisy intermediate models.

In this article, we use both solutions. The general behaviour of the algorithm we propose is to compute an expected *composite* model using Bayesian Model Averaging [2] theory and a set of expected features, the expected edge existence, computed from the learning of many *component* models. These components are themselves highly biased models, more precisely spanning arborescences [4], with the interesting properties of global optimality and polynomial time computation[5]. Due to this latter property, it is necessary to introduce variance in the dataset for each component learning task, in order to browse more of the models space. This variance is obtained here by: 1) perturbing the original dataset through sampling; 2) forbidding the use of some edges in the spanning arborescence, the edge blacklist being randomly generated for each component.

The paper is organized as follows. We describe our mixture algorithm in section 2. We then validate it in section 3 and show promising results on experiments from the popular DREAM D8C1 recent challenge [6], before discussing the algorithm and its perspectives in section 4.

2 Combining Spanning Arborescences for Network Inference

2.1 Data representation

Let us consider a matrix representation of a dataset D consisting of n ordered time stamps (D rows) over N variables (D columns). Each column j is a sequence describing an observed variable over n time steps

$\langle D_{ij} \rangle_{i \in \langle 1,\dots,n \rangle}$ and each row i describes the state of a system at time step i over the N observed variables. Our goal is to find a model of the system of interest in terms of dependencies between the observed variables at different time steps of the system. In this paper, we assume that this system is a Markov process, i.e. that each time step state only depends on the immediate previous step state, and that the transition from each state to the next state is driven by the same underlying model.

The first step for the proposed algorithm is to transform the $n \times N$ dataset into a $(n-1) \times 2N$ i.i.d. dataset (under Markov assumption) D^t describing 2 consecutive time slices of the system. The transformation consists in concatenating every pair of consecutive time steps from D into a "dynamic" example in D^t, i.e. ,we have for each row: $D^t_{i\cdot} = [D_{i,1} \, \dots \, D_{i,N} \, D_{i+1,1} \, \dots \, D_{i+1,N}]$.

2.2 Learning component models

From a dynamic dataset D^t, the first learning step of the proposed algorithm is to compute a set of *component* models, i.e. simple models which will be combined in the second part of the algorithm. Considering a number m of simple models to learn, we first compute m local perturbations of $D^t = \{D^{t[k]}\}_{1 \leq k \leq m}$ by sampling from D^t with replacement (bagging strategy). Then, for each $D^{t[k]}$, a directed graph $G^k = (V, E^k)$, with V having one vertex for each of the N original variables, is built by first randomly choosing $\alpha \cdot N \cdot (N-1)/2$ undirected edges and then computing the two directed scores $s(A \rightarrow B)$ and $s(B \rightarrow A)$ for each of them, and for a given s. Finally, each graph G^k is searched for its optimum spanning arborescence A^k with respect to the score s, using the Edmonds algorithm [4].

Even if the built graphs only have one node per original variable (as opposed to two nodes, one for timestep t and one for timestep $t+1$, the semantics of an arc $X \rightarrow Y$ measures the influence of X at time t over Y at time $t+1$. This semantics is taken into account during scores' computation.

2.3 Computing the composite model

Once the m component models have been learned, the second learning step aims at combining them into a composite model. In the Bayesian Model Averaging framework, this step is achieved by computing a set of *expected features* $\{\forall i : \mathbf{E}(f_i)\}$ for the composite model, each $\mathbf{E}(f_i)$ being inferred from each component features set $\{f^k_i\}_{1 \leq k \leq m}$. In this paper, the feature space consists in the set of all possible edges in V^2, and an expected edge score is computed by counting how often that edge was present in the arborescence A^k, considering it was present in the initial weighted graph G^k. Formally, we have for all $(A, B) \in V^2$:

$$\mathbf{E}(f_{A \rightarrow B}) \approx |\, \{k \mid (A,B) \in \textbf{edges}(A^k)\} \,| \cdot \alpha^{-1}.$$

More complex features could be considered, such as paths instead of edges or ancestor / descendant relationships, as in [2] (although the authors do not combine them in a single model). We leave these problems for future work since it would require more complex combination rules.

The computation of all edges' expected scores in the composite model directly provides a ranking for those edges which can be evaluated as is using AUROC scores.

2.4 Complexity

Time complexity can be expressed as the sum of two terms: one for the components computation, and another for the combination step. The components computation complexity is $m \cdot (s + g + e)$, where s (resp. g, e) is the complexity of sampling (resp. connected graph construction and Edmonds algorithm). The complexity of the sampling step is negligible here, but the construction of the G^k is in $\mathcal{O}(\alpha N(N-1)) \approx \mathcal{O}(N^2)$, as is the Edmonds algorithm computation with the Tarjan optimization for dense graphs [7] ($\mathcal{O}(N^2 \, log \, N)$ for sparse ones). Thus the components computation part is in $\mathcal{O}(mN^2)$.

The combination part is a succession of joins between the component edgelists for further counting. It is possible to achieve such operation in $\mathcal{O}(\sum_k |\textbf{edges}(A^k)|) \approx \mathcal{O}(mN)$ with hash joins.

Overall, the proposed approach is thus of quadratic complexity. In practice, it is also highly parallelizable, since each component learning is independent and the order of joins in the second part is not significant.

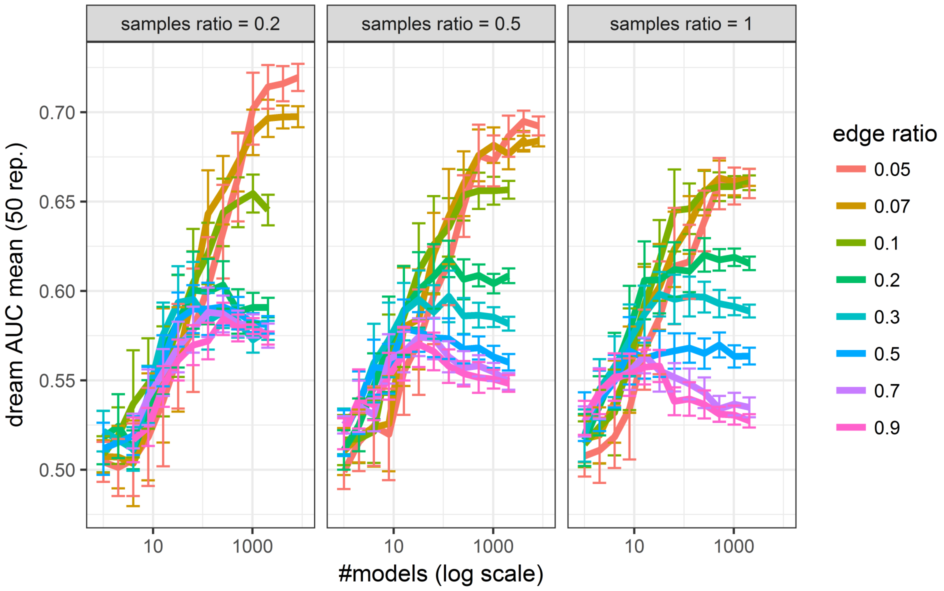

Fig. 1. Mean and sds of AUC computed by DREAMTools over 50 computations as a function of the number of combined models, as well as the samples and edge ratio used for components learning.

3 DREAM 8 (HPN-DREAM) SC1B Network Inference Challenge Results

In this section, we give preliminary encouraging results for the validation of our method and comparison with other algorithms results using a dataset of the recent HPN-DREAM 8 Breast Cancer closed challenge [6].

3.1 Challenge and evaluation method description

The DREAM 8 SC1B subchallenge learning objective is to find the network of a synthetical biological model built using state of the art methods and biological knowledge. Simulation of this model led to the production of several time series involving 20 biological features. The preprocessed dataset (cf. 2) contains 80 t to $t+1$ examples over 40 temporal features.

The evaluation of learning results for this task is achieved by an official tool, the *DREAMTools* python package [8], through the computation of an AUROC score against the golden standard. In addition to computing scores the same way from one algorithm to another, this package also provides the expected ranking an algorithm would have reached if the challenge were still open, using all final results from the official submissions (over 100), which allows for a cheap comparison with many (more or less popular) algorithms.

In order to quantify the impact of several parameters on our algorithm learning quality, we have tested the method with different parametrization of the number of combined modes m, the ratio of samples n contained in each data perturbation, and the ratio of edges α present in each graph before each component learning. The used score for edge weighting was the BDeu [5] gain between the parenting situation described by the edge, and the no parent situation; namely for an edge $A \rightarrow B$: $BDeu(parents(B) = \{A\}) - BDeu(parents(B) = \emptyset)$.

3.2 Results

Results for many parametrizations, given in Figure 1, show different clear trends. Firstly, we can see that for small edge ratios, the obtained AUC seems to monotonically increase with the number of combined models, until reaching plateaus. For bigger ratios, the trend seems to be mostly observable, but the higher the sampling ratio, the lower the minimum edge ratio needs to be to show this trend. Additionally, we can observe that the convergence AUC value tends to increase whenever any of the edge or the sample ratio decrease. These results seem to indicate that focusing on smaller parts of the available information for each component, while aggregating a higher number of them for final consensus, seem to give the best results, which confirms the requirement for components diversity in order to give a good consensus.

Concerning the expected ranking for the different results, our approach is very promising since it reached the 3^{rd} position for the best mean AUC obtained over the different parametrizations, outperforming GENIE3

[9], ARACNE [10], all heuristic oriented Bayesian network methods, as well as all linear and most non-linear regression methods, all ODE and all ensemble learning submissions.

4 Discussion

At the moment, the gap on DREAM 8 experiments between our and the best performance of the official ranking is of 0.045. A particularity of the considered DREAM challenge is that 3 out of the 20 biological features are fake nodes, supposed to have no correlation with the others. If we evaluate the models we found, after removing edges involving these nodes, we have an extra AUROC gain so that first position is reached. This shows a limitation of our approach in its current form: learning spanning arborescences means that every feature will get one parent per component, even if there is no true correlation. This problem is not trivial, since there is also a tendency for such spurious correlations to be non-uniformly distributed. Since edge samplings is uniformly done, there is a high probability for a restricted number of parents to appear in each component for a fake node.

Despite this limitation, experimental results of section 3 can help to understand the behaviour of a structure learning method based on the combination of spanning arborescences with provoked models diversity through edge and data sampling. We have first shown that combining more but more diverse models lead to better convergence values, involving in the currently defined parameter space a decrease of both sample and edge ratios. It is encouraged to use this strategy in a quite extreme way, since best performances obtained in the experiments are achieved in situations where both ratios are very low. The only warning would be to still allow the Edmonds algorithm to have choice, in order not to make the components completely random.

5 Conclusion

In this paper, we have presented a network inference learning algorithm based on the combination of multiple spanning arborescences learned over multiple perturbation of the original dataset, with enforced diversity through edge sampling, showing promising results in practice on a recent DREAM challenge. We have identified key parametrizations to improve the algorithm performances, and have discussed current limitations of the proposed method. Future works will thus consider them, in a short term, before considering more advanced extensions such as the introduction of priors, really important in biological contexts, modular capabilities, which is becoming a standard in recent methods to abstract from a model complexity, and different component combination rules to preserve extra properties in the consensus model, such as paths or path lengths.

References

[1] Ludwig Geistlinger, Gergely Csaba, Simon Dirmeier, Robert Küffner, and Ralf Zimmer. A comprehensive gene regulatory network for the diauxic shift in saccharomyces cerevisiae. *Nucleic acids research*, page gkt631, 2013.

[2] Nir Friedman, Moises Goldszmidt, and Abraham Wyner. Data analysis with bayesian networks: A bootstrap approach. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 196–205. Morgan Kaufmann Publishers Inc., 1999.

[3] Bradley M Broom, Kim-Anh Do, and Devika Subramanian. Model averaging strategies for structure learning in bayesian networks with limited data. *BMC bioinformatics*, 13(13):S10, 2012.

[4] Jack Edmonds. Optimum branchings. *Mathematics and the Decision Sciences, Part*, 1:335–345, 1968.

[5] David Heckerman, Dan Geiger, and David M Chickering. Learning bayesian networks: The combination of knowledge and statistical data. *Machine learning*, 20(3):197–243, 1995.

[6] Steven M Hill, Laura M Heiser, Thomas Cokelaer, Michael Unger, Nicole K Nesser, Daniel E Carlin, Yang Zhang, Artem Sokolov, Evan O Paull, Chris K Wong, et al. Inferring causal molecular networks: empirical assessment through a community-based effort. *Nature methods*, 13(4):310–318, 2016.

[7] Robert Endre Tarjan. Finding optimum branchings. *Networks*, 7(1):25–35, 1977.

[8] Thomas Cokelaer, Mukesh Bansal, Christopher Bare, Erhan Bilal, Brian M Bot, Elias Chaibub Neto, Federica Eduati, Alberto de la Fuente, Mehmet Gönen, Steven M Hill, et al. Dreamtools: a python package for scoring collaborative challenges. *F1000Research*, 4, 2015.

[9] Va Anh Huynh-Thu, Alexandre Irrthum, Louis Wehenkel, and Pierre Geurts. Inferring regulatory networks from expression data using tree-based methods. *PLOS ONE*, 5(9):1–10, 09 2010.

[10] Adam A Margolin, Ilya Nemenman, Katia Basso, Chris Wiggins, Gustavo Stolovitzky, Riccardo D Favera, and Andrea Califano. Aracne: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC bioinformatics*, 7(Suppl 1):S7, 2006.