

A New Local Move Operator for Reconstructing Gene Regulatory Networks

Jimmy Vandel and Simon de Givry

INRA - BIA, Toulouse, France,
jimmy.vandel@toulouse.inra.fr

Abstract. The discovery of regulatory networks is an important aspect in the post genomic research. Among structure learning approaches we are interested in local search methods in the Bayesian network framework. We propose a new local move operator to escape more efficiently from local maxima in the search space of directed acyclic graphs. This operator allows to overtake the acyclic constraint of Bayesian networks and authorizes local moves previously banned with classic operators. First results show improvements of learnt network quality. Our algorithm uses Comet language providing abstraction for local search and constraint programming.

Keywords: structure learning, Bayesian networks, local search, Comet language, gene regulation inference, genetical genomics.

1 Introduction

Inferring gene regulatory networks (GRN) from microarray data is a challenging problem, in particular because the sample size is typically small compared to the thousands of genes that compose the network. Currently, integrative approaches are developed to combine several sources of information in order to improve prediction quality. One of these approaches consists in using genetical genomic data combining gene expressions and sequence polymorphisms observed by genetic markers [1] [2](Chap. 4).

Among the many existing frameworks used to infer GRN, we choose probabilistic graphical models and more specifically *static* Bayesian Networks (BN) [3]. Learning BN structures from data is a NP-hard problem [4] and several approaches have been proposed to solve it. One of them consists in exploring the space of BN structures using local search methods and evaluating each structure with a specific scoring criterion in order to select the structure which maximizes the score.

In Section 2 we present Bayesian network and a new operator called "iterative swap cycle" (ISC) for local search algorithms. Then we report in Section 3 our preliminary work using this operator inside the Comet local search platform and give some positive results on simulated genetical genomic data.

2 Bayesian network and local search methods

A *Bayesian network* [3] denoted by $B = (\mathcal{G}, \mathbf{P}_{\mathcal{G}})$ is composed of a directed acyclic graph $\mathcal{G} = (\mathbf{X}, \mathbf{E})$ with nodes representing p random discrete variables $\mathbf{X} = \{X_1, \dots, X_p\}$,

linked by a set of directed edges \mathbf{E} , and a set of conditional probability distributions $\mathbf{P}_{\mathcal{G}} = \{P_1, \dots, P_p\}$ defined by the topology of the graph: $P_i = \mathbb{P}(X_i | Pa(X_i))$ where $Pa(X_i) = \{X_j \in \mathbf{X} | \overrightarrow{(X_j, X_i)} \in \mathbf{E}\}$ is the set of parent nodes of X_i in \mathcal{G} . A Bayesian network B represents a joint probability distribution on \mathbf{X} such that:

$$\mathbb{P}(\mathbf{X}) = \prod_{i=1}^p \mathbb{P}(X_i | Pa(X_i)) \quad (1)$$

The conditional probability distributions $\mathbf{P}_{\mathcal{G}}$ are determined by a set of parameters, θ , via the equation:

$$\mathbb{P}(X_i = k | Pa(X_i) = j) = \theta_{ijk}$$

where k is a value of X_i , and j is a value configuration of the parent set $Pa(X_i)$. Given the structure G , parameters θ_{ijk} can be estimated by following the maximum likelihood principle.

Learning the structure of a Bayesian network consists in finding a DAG \mathcal{G} maximizing $\mathbb{P}(\mathcal{G} | \mathbf{D})$ where \mathbf{D} represents the observed data. We use in our study the popular Bayesian Dirichlet criterion to maximize the score:

$$BDeu(\mathcal{G}) = \prod_{i=1}^p \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(n_{ij} + \alpha_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(n_{ijk} + \alpha_{ijk})}{\Gamma(\alpha_{ijk})}$$

with n_{ijk} , the number of occurrences of the configuration $(X_i = k, Pa(X_i) = j)$ in the n samples, $n_{ij} = \sum_{k=1}^{r_i} n_{ijk}$ and Dirichlet hyper-parameters $\alpha_{ijk} = \frac{\alpha}{r_i * q_i}$ where α is the *equivalent sample size* parameter, r_i is the domain size of variable X_i and $q_i = \prod_{X_j \in Pa(X_i)} r_j$, is the product of the parental domains of X_i .

In a GRN context with genetical genomic data the set of discrete random variables \mathbf{X} is composed of one variable per gene-activity, denoted G_i , and one variable for each genetic marker, denoted M_i , $\forall i \in \{1, \dots, p\}$ with p the number of genes. We assume each gene G_i is co-located with a single genetic marker M_i . Each marker may explain the variation of its associated gene activity or the variations of other regulated genes. An example is given in Figure 1.

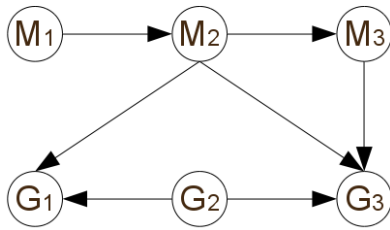


Fig. 1. Example of 3-genes network with regulations from gene 2 to gene 1 and 3. We assume a marker order (M_1, M_2, M_3) on a single chromosome.

Heuristic local search algorithms are widely used to learn Bayesian network structures as hill-climbing search, simulated-annealing, MCMC, genetic algorithms, ant colony optimization [5] and dozens more with additional refinements [6]. These methods are

often compared in previous papers with different datasets and results but if these methods tend to develop sophisticated algorithms to select at each step the best neighbor, only few of them tried new local operators to define this neighborhood [7, 8, 9]. Other approaches working on larger neighborhoods collapse a set of DAGs into a unique representative configuration. For instance, they explore the search space of total variable orderings (an optimal DAG compatible with the order is then easier to deduce) [10, 11], or the search space of Markov-equivalent partially-oriented DAGs [12, 13, 14]. Classical operators are addition, deletion and reversal of a directed edge, but these operators lead to reach quickly local maxima, even if some metaheuristic principles like Tabu list or simulated-annealing reduce this drawback. Furthermore the acyclic constraint of Bayesian networks is often considered as a hard constraint to define the neighborhood of a graph. We propose a new operator called ISC (Iterative Swap Cycle) to potentially overcome this constraint.

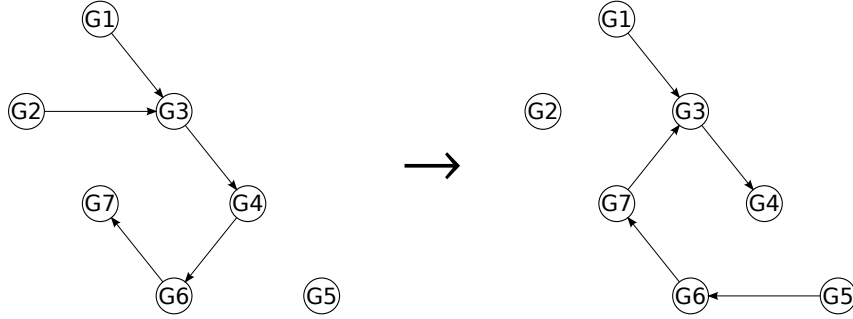


Fig. 2. Modification of 7-gene network structure after an ISC operation.

Let us consider the situation in Figure 2. We call Δ_{G_i, G_j} the BDeu score variation when we add variable G_i as parent of G_j with other parents fixed, this variation is positive if the score increases when we add the arc $G_i \rightarrow G_j$ and negative otherwise. We define the swap operator as follow, swapping an edge $G_i \rightarrow G_j$ with G_k produces the simultaneous deletion of $G_i \rightarrow G_j$ and the addition of $G_k \rightarrow G_j$. Given the initial structure in Figure 2, to swap $G_2 \rightarrow G_3$ with G_7 will be forbidden with classical operators because of the acyclic constraint even if Δ_{G_7, G_3} is high. The idea of ISC operator is to add $G_7 \rightarrow G_3$ anyway and if needed to break the cycle created by this addition. First we remove an edge of this cycle minimizing Δ value (in this example we assume $G_4 \rightarrow G_6$), if this deletion plus the swap operation do not decrease the score of the graph ($\Delta_{G_7, G_3} - \Delta_{G_2, G_3} - \Delta_{G_4, G_6} > 0$) the move is validated. Otherwise we choose another parent (not included in the considered cycle) for node G_6 maximizing Δ value (here we add $G_5 \rightarrow G_6$) that we can consider as a swap of $G_4 \rightarrow G_6$ with G_5 . If another cycle is created during this swap, we iterate a new cycle break operation. Finally we iterate until the initial cycle is broken (means no sub-cycles were created) or the score of the modified graph cannot be greater than the initial score. We only validate local moves if all the cycles are broken and if the modified graph increases the score.

If several cycles are created by the same edge addition, we break each of them, one after another by applying ISC operator for each one.

The main idea of ISC operator is to try, each time we want to add a forbidden edge $G_i \rightarrow G_j$, to break the cycles by deleting or swapping a parent for one node of the cycles and to iterate this operation to solve potential cycles created during the swap. A forbidden edge could appear when we try to swap an edge as in our previous example but also after an addition or reversal of an edge. So ISC operator is applicable for add, reverse and swap operators.

3 On going work

We use the Comet software [15] to implement iterated hill-climbing search. Comet is a specific language which provides useful concepts for implementing local search methods like invariant, objective and constraint functions. In our implementation we encode BDeu score using cache for up to two parents and incremental cycle detection (using incremental topological ordering [16]) using user-defined invariants on a graph which means that each modification on the graph is automatically propagates on the score and cyclicity test, allowing an easier way to develop new neighborhood operators and new search algorithms. We do not use any constraint features of Comet but invariants in Comet offer incrementality for free. For each node we keep in memory the neighborhood defined by all operators applicable on this node with associated score variations and potential cyclic situations, which saves computations and helps to quickly update cyclic situations.

We did not implement ISC operator yet, but we developed its simplified version called nISC (non Iterative Swap Cycle). nISC is similar to ISC but does not iterate on potential cycle creations, it try to delete or swap only one edge of the cycle produced by classical operator to break it. If deleting or swapping an edge does not break the cycle or create another cycle, the classical operation is tagged as invalid and cannot be applied in this configuration. This operator is more simple and less time consuming than ISC but allows to overcome some cycle situations. We show in Figure 3 BDeu scores reached (with $\alpha = 1$) and time requirement of 1000 runs of hill climbing search starting from a random structure (2 randomly selected parents per node) in five configurations. These configurations differ by the set of authorized operators during the search among classical ones (A:addition, D:deletion and R:reversal) and the swap operator (S:swap). The last configuration represents the results with nISC extension (*:nISC) for addition, reversal and swap operators (deletion of an edge cannot create a cycle). Our test network is composed of 2 000 nodes (1 000 genes and 1 000 markers) and 300 samples from DREAM5 challenge (SysGenA300_Network1) [17]. In order to deal with large datasets, we restrict the list of candidate parents as done in [18].

We see in Figure 3 that the scores increase as more operators are used but in counterpart slows down the search.

If the reversal operation increases a little the mean score, we see a significant improvement when we use the swap operator. These results show that swapping an edge allows a deeper structure modification than reversing it although both are composed of an addition and a deletion. Furthermore variance of scores is reduced with the swap

operator. Efficiency of the swap operator is simply explained by the fact that initial structures could allocate for one node G_j , a medium quality parent G_i ($\Delta_{G_i, G_j} > 0$). However if a better parent G_k exists for this node $\Delta_{G_k, G_j} > \Delta_{G_i, G_j}$ but both G_i and G_k cannot be parents at the same time we would need to swap G_i by G_k . So we first need to remove G_i but this operation will decrease the score and will not be considered in a hill-climbing search or with low probability in a simulated-annealing algorithm.

Applying nISC with the four operators allows to increase once more the mean score reached and to reduce variance which suggest that trying to overcome cycles allows to escape from local optima more and to achieve similar quality structures. But this more intensive search is much more time consuming even if the current implementation could be improved. Trade off between search time and quality of the learnt structure still need to be investigated.

	A+D	A+D+R	A+D+S	A+D+R+S	A*+D+R*+S*
BDeu scores (in \log_{10})					
mean	-360 415	-360 349	-358 885	-358 826	-358 417
variance (10^3)	25.936	26.681	4.446	4.763	2.713
Mean Time (seconds)	22.2	28.2	36.6	40.2	153

Fig. 3.

Our first results show large difficulty for local search algorithms to escape from local maxima with classical operators. A difficulty which can explain poor results we obtained with the simulated annealing method [19], even if a progressive decrease of the temperature can move the search in a promising area, but when temperatures become too low the algorithm quickly suffers from restrictive operators and falls in a local maximum. For this reason instead of developing highly complex metaheuristics, we decide to explore new operators to define larger neighborhoods.

References

- [1] R. Jansen and J. Nap, "Genetical genomics : the added value from segregation," *Trends in genetics*, vol. 17, no. 7, pp. 388–391, 2001.
- [2] S. Das, D. Caragea, W. H. Hsu, and S. M. Welch, eds., *Handbook of Research on Computational Methodologies in Gene Regulatory Networks*. Hershey, New York: IGI Global, 2010.
- [3] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.
- [4] D. Chickering and D. Heckermann, "Learning bayesian networks is NP-complete," *In learning from data: AI and Statistics*, 1996.
- [5] R. Daly and Q. Shen, "Learning bayesian network equivalence classes with ant colony optimization," *Journal of Artificial Intelligence Research*, vol. 35, pp. 391–447, 2009.
- [6] E. Salehi and R. Gras, "An empirical comparison of the efficiency of several local search heuristics algorithms for bayesian network structure learning," in *Learning and Intelligent Optimization Workshop (LION 3)*, 2009.

- [7] L. de Campos, J. Fernandez-Luna, and J. Puerta, "Local search methods for learning bayesian networks using a modified neighborhood in the space of dags," in *Advances in Artificial Intelligence IBERAMIA 2002*, vol. 2527, pp. 182–192, 2002.
- [8] A. Moore and W. Wong, "Optimal reinsertion: A new search operator for accelerated and more accurate bayesian network structure learning," in *Proceedings of the 20th International Conference on Machine Learning (ICML '03)*, pp. 552–559, AAAI Press, 2003.
- [9] A. Holland, M. Fathi, M. Abramovici, and M. Neubach, "Competing fusion for bayesian applications," in *Proc. of the 12th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2008)*, pp. 378–385, 2008.
- [10] G. Cooper and E. Hersovits, "A bayesian method for the induction of probabilistic networks from data," *Machine Learning*, vol. 9, pp. 309–347, 1992.
- [11] M. Teysier and D. Koller, "Ordering-based search: A simple and effective algorithm for learning bayesian networks," in *Proceedings of the Twenty-first Conference on Uncertainty in AI (UAI)*, pp. 584–590, 2005.
- [12] D. Chickering and D. Maxwell, "Learning equivalence classes of bayesian-network structures," *J. Mach. Learn. Res.*, vol. 2, pp. 445–498, 2002.
- [13] S. Acid and L. M. de Campos, "Searching for bayesian network structures in the space of restricted acyclic partially directed graphs," *J. Artif. Int. Res.*, vol. 18, pp. 445–490, 2003.
- [14] D. Fierens, J. Ramon, M. Bruynooghe, and H. Blockeel, "Learning directed probabilistic logical models: ordering-search versus structure-search," *Annals of Mathematics and Artificial Intelligence*, vol. 54, pp. 99–133, 2008.
- [15] L. Michel and P. V. Hentenryck, "A constrained-based architecture for local search," in *17th Annual ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications*, 2002.
- [16] D. J. Pearce and P. H. J. Kelly, "A dynamic topological sort algorithm for directed acyclic graphs," *J. Exp. Algorithmics*, vol. 11, 2007.
- [17] "Dream5, systems genetics challenges." <http://wiki.c2b2.columbia.edu/dream/index.php/D5c3>, 2010.
- [18] A. Goldenberg and A. Moore, "Tractable learning of large bayes net structures from sparse data," in *Proceedings of the twenty-first international conference on Machine learning*, ICML '04, pp. 44–51, 2004.
- [19] D. Chickering, D. Heckerman, and D. Geiger, "Learning bayesian networks: search methods and experimental results," *AI + Stats*, 1995.