Weighted Committee-Based Structure Learning for Microarray Data

Hasna Njah and Salma Jamoussi

***Abstract*—Bayesian networks (BN) are considered to be one of the strongest modeling techniques of gene regulatory networks (GRN) thanks to their ability to present features and relations between them in a causal and probabilistic way. Learning the structure of those models needs a large training dataset in order to avoid over-fitting. However, biological data, especially microarray data, suffer from the presence of only few instances. Some recent approaches tried to face this challenge by applying committee based methods. We use this principle in order to suggest a new method supported by a double-weight-assignment technique. We show that our approach has succeeded to learn benchmark structures.**

***Index Terms*—Bayesian Network, Structure Learning, Committee Learning, Gene Regulatory Networks**

I. Introduction

BAYESIAN network (BN) is a powerful Gene Regulatory Network (GRN) modeling technique. It plays an important role in determining regulatory relations between genes implicated in a disease. It allows presenting biological relations between features (genes) through a causal and probabilistic model.

As it is well known, BN consists of a directed acyclic graph (DAG) and nodes corresponding probability tables (CPTs). BN learning requires two major steps: structure learning (SL), which allows establishing the DAG, and parameter learning, which compute values of CPTs.

In literature, there are various SL techniques which differ on how they search the optimal DAG. They are divided into three major families: constraint-based methods, search-and-score based methods and hybrid methods. Constraint-based approaches aim to find conditional independencies using independencies test, such as Chi-squared test, and to construct a final graph basing on this knowledge. Examples of these approaches are IC (Inductive Causality) [1], PC [2], CI (Conditional Independence) [3], etc. Search-and-score based approaches use likelihood-based score as an alternative to statistical independency tests. These scores, which aim to estimate the quality of a BN, are to be optimized in DAG space. Some of the most used scores are BDeu (Bayesian Dirichlet equivalent uniform) [4] and AIC (Akaikes Information Criterion) [5]. Search-and-score based approaches can be grouped into different families. For instance, we distinguish deterministic and stochastic approaches. Deterministic approaches have approximately the same result when using different parameters such as K2 [6], Tabu [7] and Hill-Climbing [8]. Stochastic approaches can provide different graphs for the same input parameters like Simulated Annealing [9] and Metropolis-Hastings algorithm [10]. Search-and-score based approaches can be optimized by restricting the space of possible structures. For instance, tree-based approaches like Chow-Liu Tree [11] build the Maximal Weight Spanning Tree (MWST) of an obtained graph. Finally, hybrid approaches aim to benefit from constraint-based and search-and-score based approaches. They apply a local search to find independencies in a neighborhood and a global search to find the optimized DAG. An example of these methods is Max-Min-Hill-Climbing (MMHC)[12].

Despite the variety of its algorithms, SL step faces two major challenges in the context of high dimensional data (e.g., microarray data) which are treating the gigantic number of features and overcoming the limited number of instances. In fact, SL algorithms are NP-complete [13] because the number of possible DAGs grows exhaustively with a small augmentation of nodes (features) number. In that way, searching the optimal structure is quite impossible when considering a few dozens of features. A possible solution is to divide the training dataset into small partitions using cluster analysis techniques. Resultant BNs are linked to each other to form the final DAG that presents relations between features. However, the veracity of these relations is not ensured because, generally, BNs tend to over-fit the training data, when trained on a small dataset [14]. Our contribution is to apply a committee based SL, where the final structure is learnt by a weighted vote. This weight differs by the used training set.

II. Related Work

Researchers attempt to overcome the problem of over-fitting by applying different specialized SL techniques such as bootstrap approach, evolutionary algorithms and set-based techniques.

Bootstrap approaches are founded on the regeneration of sub-sets of training data and searching for the best structure. They are widely applied to learn structure from limited data [15], [16].

Evolutionary Algorithms (EA), which are inspired from biological evolution, have demonstrated their effectiveness in case of limited number of instances[17], [18]. For instance, Genetic Algorithms present an optimized heuristic for SL. They are based on the evolution of partial candidate structures and the maximization of fitness measure [19]. Moreover, an

Hasna Njah is with Multimedia InfoRmation Systems and Advanced Computer Laboratory (MIRACL), Sfax University, Tunisia, e-mail: hasna.njah.tn@ieee.org.

Salma Jamoussi is with Multimedia InfoRmation Systems and Advanced Computer Laboratory (MIRACL), Sfax University, Tunisia, e-mail: salma.jamoussi@isimsf.rnu.tn.

978-1-4799-3163-7/13/$31.00 ©2013 IEEE

example of a combination between genetic and evolutionary SL algorithms is found in [20].

Set based SL techniques try to escape the over-training problem and ensuring an improved quality of the final structure by applying committee-based SL [21]. For instance, [22] uses a fast committee based SL algorithm on a given neighborhood. It determines the orientation of edges by the use of a majority vote. Another set-based SL technique is proposed in [23]. The idea of this approach is to apply a voting procedure for globally learning the structure of a BN.

III. Proposed Approach

Our approach is composed of two major phases which are multiple SL and BN fusion. Multiple SL phase is characterized by choosing a training dataset and weighting each one of the committee members according to their performance in classifying the instances of the chosen dataset. We use cross-validation on the training set with ten folds. During the BN fusion phase, the learnt structures issued form each one of the committee members is object to a boosting operation in order to find the best structure. The architecture of our approach is explicated in figure 1.

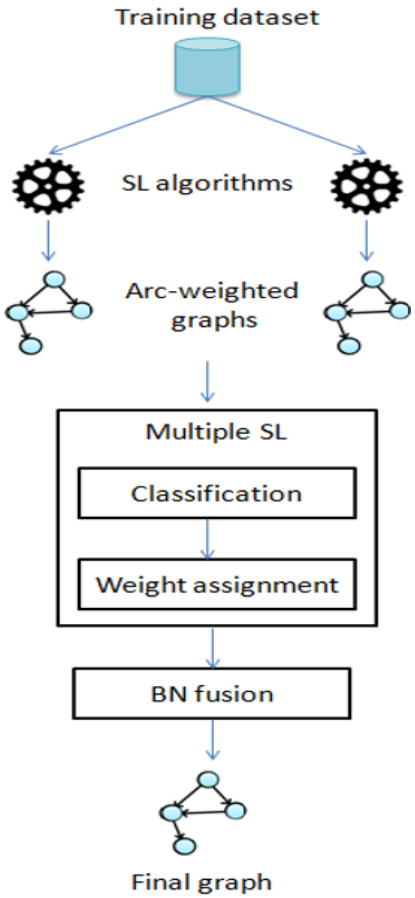

Figure 1. The major steps of our approach

A. Committee members

In this work, we chose to establish a diversified committee composed of the most known existing algorithms in each one of structure learning families. The committee members are:

- PC algorithm: it uses Chi-square test (X2) to evaluate the existence of conditional independency between two features.
- Tabu search: it defines the neighbor of a current state of the graph and chooses the best structure basing on its score. It uses a tabu list to save visited structures.
- Simulated Annealing (SA): it searches the best structures by minimizing a temperature function. A greedy search is applied to find the optimal one.
- Chow Liu Tree algorithm (CLT): it finds the best tree representation of the joint distribution induced by the data. It fixes weight to arcs basing on mutual information between features, defines the MWST over all features and direct arcs.
- MMHC algorithm: this hybrid algorithm uses constraint identification to create an undirected graph and uses this graph to apply a HC search for the final structure.

B. Weight Assignment

We used two weight assignment procedures in two different contexts. The first one is for arcs importance in the graph and the second one is for the algorithms performance. For the former, there is multitude of arcs strength assessing methods which differ basing on the chosen criterion [24]. For instance, the strength of an arc can be measured by a conditional independence test such as a p-value, so the lower value corresponds to the strongest arc. Another possible method is to find the strength of all possible arcs as learned from bootstrapped data. Also, weighting arcs can be measured by a score function. Therefore, arcs strength is defined as the gain/loss that affects the BN when removing this arc. We applied the last method on weighting arcs in our approach. Thus, the arcs weight in a model learnt by an algorithm, a, from the committee can handle three possible values.

$$Warc_a(i,j) = \begin{cases} 0 & if\ there\ is\ no\ arc\ from\ node\ i\ to\ node\ j \\ 0.5 & if\ there\ is\ a\ strong\ arc\ from\ node\ i\ to\ node\ j \\ 1 & if\ there\ is\ a\ weak\ arc\ from\ node\ i\ to\ node\ j \end{cases} \quad (1)$$

For the latter, algorithms are weighted according to their ability to bi-classify instances. So, during this task, we consider committee members as classifiers rather than SL algorithms. To do so, there are plenty of multi-criteria metrics, in literature, such as accuracy, recall, precision, true negative rate, etc.

Each one of those metrics treats a particular point of view for evaluating a SL algorithm.

For instance, recall (2) measures the classifiers ability to class positive instances as such and precision (3) measures its ability to classify negative instances as such.

$$R = \frac{TP}{TP + FN} \tag{2}$$

and

$$P = \frac{TP}{TP + FP} \tag{3}$$

where TP (True Positives) = correctly identified instances, FP (False Positives) = incorrectly identified instances and FN (False Negatives) = incorrectly rejected instances.

In this work, we focused on the ability that the learnt structure accepts all relevant solutions (class all positives as such and negatives as such) and reject others.

For this reason, we use the F-measure (4), which is the harmonic mean of specificity and sensitivity, to evaluate each algorithms performance among a classification task vis-à-vis the same training dataset.

$$F(a) = \frac{2 \times R \times P}{R + P} \tag{4}$$

Furthermore, we use the f-measure to elaborate a weight assignment formula (5).

Each algorithm a_i from the committee A receives w weight $W_{algo}(a_i)$ computed as follow:

$$W_{algo}(a_i) = \frac{(F(a_i))}{(\sum_{(} a_j \in A) F(a_j))} \tag{5}$$

C. Fusion of Learnt Strctures

Having algorithms weights and arcs weights for each algorithm, as input, it becomes possible to establish the partial fusion matrix PF.

$$PF(i,j) = \sum_{a_j \in A} W_{algo}(a_i) \times Warc_{a_i}(i,j) \tag{6}$$

We considered this matrix as partial fusion matrix because it may contain two different values of the same arc depending on its directionality i.e. $PF(i,j) \neq PF(j,i) \neq 0$. For this reason, we apply a directionality determination algorithm (1) .

Algorithm 1 Directionality determination algorithm

1. input: PF matrix
2. n: the number of features
3. for each $i = 1..n$ do
4. for each $j = 1..i-1$do
5. if $PF(i,j) \geq PF(j,i)$ then
6. $DPF(i,j) \leftarrow PF(i,j) + PF(j,i)$
7. else
8. $DPF(j,i) \leftarrow PF(i,j) + PF(j,i)$
9. return: DPF matrix

The DPF matrix is triangular with no values on the diagonal. It contains the mean of directed partial fusion matrix.

$$FM = \frac{1}{c} DPF \tag{7}$$

Where c denotes the cardinality of committee members ensemble. In our case, it is equal to five. We finalize the fusion step by applying a thresholding to the final arcs.

We keep only the arcs whose weight is greater than $W_{algo}(a_s)$. Where a_s is the SL algorithm in the committee with the highest weight.

IV. Experimental Results

We tested our approach on several dataset with known generated structure. We limited the number of instances to 100 for each dataset so that we can evaluate our methods performance face to limited instances challenge.

We present results obtained when using ASIA dataset (8 attributes), HEART dataset (15 attributes) and INSURANCE dataset (27 attributes). Table I shows the weights assigned to each one of the committee members when executed on the same dataset.

Algorithm	Asia	Heart	Insurance
PC	0.1	0.16	0.10
Tabu Search	0.3	0.36	0.18
SA	0.2	0.08	0.18
CLT	0.2	0.16	0.29
MMHC	0.2	0.24	0.25

Table I
WEIGHTS OF COMMITTEE MEMBERS ACCORDING TO THE EACH DATASET.

We apply our approach on these datasets to find the final structures. The learnt structures found by our method are evaluated to the known structure.

To do so, we used a confusion matrix for each dataset where:

TP (True Positives) = arcs correctly accepted by our system

FP (False Positives) = arcs incorrectly accepted by our system

FN (False Negatives) = arcs incorrectly rejected by our system

We measure the Euclidian distance between the graph of weighted fusion G_f and the known graph G_0 following this formula 8

$$Dist(G_f, G_0) = \sqrt{(t - TP)^2 + FP^2 + FN^2} \tag{8}$$

where t is the number of arcs in G_0.

We measure the same distance (8) between the graph of non weighted fusion Gn and the known graph G_0 : $Dist(G_n, G_0)$. We apply this approach on the three benchmarks. For each benchmark, we present the Euclidian Distance measures computed between the known structure G_0 and the learnt BN G_f and the known structure G_0 and the non-weighted learnt structure G_n in Table II.

	Asia		Heart		Insurance	
	G_f	G_n	G_f	G_n	G_f	G_n
$Dist(G_0,.)$	2	4	4.24	10	11.22	30.43
t	8		12		52	

Table II
MEASURES OF DISTANCE BETWEEN THE KNOWN GRAPH AND LEARNT GRAPHS

Table 2 shows the efficiency of using a weighted committee SL in order to learn a close BN to the real one using a limited number of instances in the training set.

V. Conclusions

The proposed approach showed its great performance when applied to benchmarks. It is able to find a close approximation to the real structure even when using a small training dataset. As a future work, we opt to apply this approach on microarray databases in order to identify regulatory relations between key disease genes. We attempt to propose a better weighting method either for algorithms or for arcs. When progressively improved, our approach can lead to a useful assistance to biologists for the quest of finding relations between genes implied in a given disease.

References

[1] J. Pearl, *Causality Models, Reasoning, and Inference*. MIT Press, 2000.

[2] P. Spirtes, C. Glymour, and R. Scheines, *Causation, Prediction, and Search*. MIT Press, 2001.

[3] J. Cheng, R. Greiner, J. Kelly, D. Bell, and W. Liu, "Learning bayesian networks from data: An information-theory based approach," *Artificial Intelligence*, vol. 137, pp. 43 – 90, 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0004370202001911

[4] Y. M. Shtarkov, "Universal sequential coding of single messages," *Problems of Information Transmission*, vol. 23, no. 3, pp. 175–186, 1987.

[5] H. Akaike, "A new look at the statistical model identification," *Automatic Control, IEEE Transactions on*, vol. 19, no. 6, pp. 716–723, 1974.

[6] G. Cooper and E. Herskovits, "A bayesian method for the induction of probabilistic networks from data," *Machine Learning*, vol. 9, no. 4, pp. 309–347, 1992. [Online]. Available: http://dx.doi.org/10.1007/BF00994110

[7] F. Glover and M. Laguna, *Tabu search, in Modern Heuristic Techniques for Combinatorial Problems,*, C. Reeves, Ed. Blackwell Scientific Publishing, Oxford, 1993.

[8] Cormen, Leiserson, and Rivest, *Introduction to Algorithms*. Cambridge Mass.: MIT Press, 1990.

[9] M. Janđura and J. Nielsen, "A simulated annealing-based method for learning bayesian networks from statistical data," *International Journal of Intelligent Systems*, vol. 21, no. 3, pp. 335–348, 2006. [Online]. Available: http://dx.doi.org/10.1002/int.20138

[10] D. Madigan, J. York, and D. Allard, "Bayesian graphical models for discrete data," *International Statistical Review / Revue Internationale de Statistique*, vol. 63, no. 2, pp. 215–232, 1995. [Online]. Available: http://www.jstor.org/stable/1403615

[11] C. I. Chow, S. Member, and C. N. Liu, "Approximating discrete probability distributions with dependence trees," *IEEE Transactions on Information Theory*, vol. 14, pp. 462–467, 1968.

[12] I. Tsamardinos, L. E. Brown, and C. F. Aliferis, "The max-min hill-climbing bayesian network structure learning algorithm," *Mach. Learn.*, vol. 65, no. 1, pp. 31–78, Oct. 2006. [Online]. Available: http://dx.doi.org/10.1007/s10994-006-6889-7

[13] D. M. Chickering, "Learning bayesian networks is np-complete," 1996.

[14] G. Elidan and S. Gould, "Learning bounded treewidth bayesian networks," in *Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 8-11, 2008*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds. MIT Press, 2008, pp. 417–424.

[15] E. Prestat, S. R. de Morais, J. A. Vendrell, A. Thollet, C. Gautier, P. A. Cohen, and A. Aussem, "Learning the local bayesian network structure around the {ZNF217} oncogene in breast tumours," *Computers in Biology and Medicine*, vol. 43, no. 4, pp. 334 – 341, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0010482512002065

[16] A. Djebbari and J. Quackenbush, "Seeded bayesian networks: Constructing genetic networks from microarray data," *BMC Systems Biology*, vol. 2, no. 1, p. 57, 2008. [Online]. Available: http://www.biomedcentral.com/1752-0509/2/57

[17] L. Wang, X. Wang, A. P. Arkin, and M. S. Samoilov, "Inference of gene regulatory networks from genome-wide knockout fitness data," *Bioinformatics*, vol. 29, no. 3, pp. 338–346, Feb. 2013. [Online]. Available: http://dx.doi.org/10.1093/bioinformatics/bts634

[18] C. Auliac, F. d'AlchBuc, and V. Frouin, "Learning transcriptional regulatory networks with evolutionary algorithms enhanced with niching," in *Applications of Fuzzy Sets Theory*, ser. Lecture Notes in Computer Science, F. Masulli, S. Mitra, and G. Pasi, Eds. Springer Berlin Heidelberg, 2007, vol. 4578, pp. 612–619. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-73400-0_78

[19] E. K. J. Lee, W. Chung and S. Kim, "A new genetic approach for structure learning of bayesian networks : Matrix genetic algorithm," *International Journal of Control, Automation and Systems*, vol. 8, p. 398 407, 2010.

[20] A. Carvalho, "A cooperative coevolutionary genetic algorithm for learning bayesian network structures," in *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, ser. GECCO '11. New York, NY, USA: ACM, 2011, pp. 1131–1138. [Online]. Available: http://doi.acm.org/10.1145/2001576.2001729

[21] S. Stahlschmidt, H. Tausendteufel, and W. K. Hrdle, "Bayesian networks for sex-related homicides: structure learning and prediction," *Journal of Applied Statistics*, vol. 40, no. 6, pp. 1155–1171, 2013.

[22] E. Mwebaze and J. A. Quinn, "Fast committee-based structure learning." *Journal of Machine Learning Research - Proceedings Track*, vol. 6, pp. 203–214, 2010. [Online]. Available: http://dblp.uni-trier.de/db/journals/jmlr/jmlrp6.html#MwebazeQ10

[23] K. Abu-Hakmeh, "Assessing the use of voting methods to improve bayesian network structure learning," Ph.D. dissertation, Georgia Institute of Technology, December 2012.

[24] R. Nagarajan, M. Scutari, and S. Lbre, *Bayesian Networks in R: with Applications in Systems Biology*. Springer Publishing Company, Incorporated, 2013.