A Multi-parent Search Operator for Bayesian Network Building

David Iclănzan

Department of Computer Science, Babe s-Bolyai University,
Kogălniceanu no. 1, 400084, Cluj-Napoca, Romania
david.iclanzan@gmail.com

Abstract. Learning a Bayesian network structure from data is a well-motivated but computationally hard task, especially for problems exhibiting synergic multivariate interactions. In this paper, a novel search method for structure learning of a Bayesian networks from binary data is proposed. The proposed method applies an entropy distillation operation over bounded groups of variables. A bias from the expected increase in randomness signals an underlaying statistical dependence between the inputs. The detected higher-order dependencies are used to connect linked attributes in the Bayesian network in a single step.

1 Introduction

A Bayesian networks is a probabilistic graphical model that depicts a set of random variables and their conditional independence via a directed acyclic graph. It represents a factorization of a multivariate probability distribution that results from an application of the product theorem of probability theory and a simplification of the factors achieved by exploiting conditional independence statements of the form $P(A|B, X) = P(A|X)$, where A and B are attributes and X is a set of attributes.

The represented joint distribution is given by:

$$P(A_1, \ldots, A_n) = \prod_{i=1}^{n} P(A_i|par(A_i)) \tag{1}$$

where $par(A_i)$ denotes the set of parents of attribute A_i in the directed acyclic graph that is used to represents the factorization.

Bayesian networks provide excellent means to structure complex domains and to draw inferences. They can be acquired from data or be constructed manually by domain experts (a tedious and time-consuming task).

One of the most challenging task in dealing with Bayesian networks is learning their structures, which is an NP-hard problem [1,2]. Most algorithms for the task of automated network building from data, consist of two ingredients: a search method that generates alternative structures and an evaluation measure or scoring function to assess the quality of a given network by calculating the goodness-of-fit of a structure to the data.

C.A. Coello Coello et al. (Eds.): PPSN 2012, Part I, LNCS 7491, pp. 246–255, 2012.

Due to the computational cost implications [2], most of the algorithms that learn Bayesian network structures from data use a heuristic local search to find a good model, trading accuracy for tractability and efficiency. Exact Bayesian network learning has a $O(n2^n)$ complexity, thus it is feasible up to 30 variables [3].

Heuristic methods, at each step apply some search operators like perturbation or solution mixing, to some current network structure(s), exploring their neighborhoods. After evaluating the new solutions, they promote changes that result in the improvement of some discriminative metric.

Because these search methods alter only a few arcs at the time, they can hardly find and express multivariate interactions that only manifest at a synergic level like the parity function. Here, adding edges between less than k nodes, where k is the size of the block containing the multivariate interaction, will not result in any improvement, thus are hard to discover by methods closely following the discriminative metric gradient.

In this paper we propose a linkage-detection method that is able to select all relevant parents for an attribute in one step, by finding and expressing even relationships not manifesting at pairwise level. Our method exploits the property of the exclusive or (XOR) operator to produce randomness from non-deterministic sources. We search for groups of variables where entropy distillation does not occur, signaling a non-determinism in the source - statistical dependence between the variables.

Albeit a costly search for the groups of variables must be performed, this approach enables the correct detection of Bayesian network, unattainable by simple heuristic search methods.

2 Detecting Higher-Order Dependencies

Binary problems of real interest may have many variables with complicated multivariate interactions among them. The dependency of a binary variable X_e on a (noisy) feature expressed by several other variables of the problem can be formalized as follows:

$$\begin{aligned} \text{if } \ f_b(X_{v1}, X_{v2}, \ldots, X_{vl}) \ \ & [\text{and } noise(X)] \\ & \text{then} \qquad\qquad X_e = b \\ & [\text{else} \qquad\qquad X_e = \overline{b}] \end{aligned}$$

where f_b is an arbitrary deterministic boolean function of l binary variables, which analyzes if the input variables satisfy a certain feature or not. $\overline{b}$ is the bitwise negation of b. As the relation must not be fully specified, the else branch is optional. The optional boolean $noise(X)$ function can be used to introduce stochasticity to the relation, to model external influences or factors which are not directly considered when evaluating the feature. This boolean function may prevent the expression of the feature even if the conditions are present, thus adding noise to the relation.

For these kind of problems, pairwise dependencies might be very small or lacking altogether. Therefore, finding the correct dependency structure is a very hard task. Prospective methods must combine an extensive higher order model search guided by a criteria that evaluates the quality of the model in rapport with the evidence, like the Minimum Description Length (MDL) principle [4] or Bayesian-Dirichlet metric [5].

The complexity of the model determination is a product of the complexity of the search and candidate model evaluation.

For problems where statistical dependence can only be detected by considering at least k variables, the search must enumerate at least all combinations of variables taken k at the time.

An ordered tuple of binary random variables $X = (X_1, X_2, \ldots, X_n)$ is independent iff the joint distribution $Pr(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n)$ and the product of the independent ones $\prod_{k=1}^{n} Pr(X_k = x_k)$ is equal for all $x = (x_1, x_2, \ldots, x_n) \in \{0, 1\}^n$.

Relative entropy or Kullback-Leibler divergence [6] can be used to measure the "distance" between these two distributions:

$$D_{KL}(p||q) = \sum_{x \in X} p(x) \log_2 (\frac{p(x)}{q(x)}) \tag{2}$$

Measuring the D_{KL} between the observed joint distribution of some variables and the product of independent joint distribution one can measure the information gain by considering a group of variables linked. The complexity of calculating D_{KL} is exponential in k with a base equal to the cardinality of the random variables. Thus, for the binary case the complexity is $O(2^k)$.

While the burden of the combinatorial search can not be obviated, in the following we consider ways in which the complexity of the model discriminatory function can be heavily reduced from the exponential complexity.

3 Entropy Distillation Based Multivariate Dependency Detection

Most practical sources of randomness, be it hardware or pseudo-random number generators, exhibit a certain level of imperfection or bias. A perfectly random bit has an entropy of one bit and bias of 0. To obtain a highly random bit, there are algorithms that combine multiple, streams of imperfect random bits, each with entropy less than one, to create a single bit with entropy one and bias 0. This process is called entropy distillation or entropy extraction.

Exclusive OR (XOR, also denoted by $\otimes$) is commonly used to reduce the bias from imperfectly random bits, provided that the random bits are statistically independent.

The reduction in bias by repeatedly applying the XOR on non-deterministic inputs can be computed using the Piling-Up Lemma [7].

Lemma 3.1. *Let X_i for $i \in \overline{1,n}$ be statistically independent random binary variables, where p_i is the probability that $X_i = 0$ and $\epsilon_i = p_i - 1/2$ are the biases. Then the probability that $X_1 \otimes X_2 \otimes \ldots \otimes X_n = 0$ is*

$$\frac{1}{2} + 2^{n-1} \prod_{i=1}^{n} \epsilon_i \tag{3}$$

Note that as biases $\epsilon_i \in [0, 1]$ their product is a monotonically decreasing function. If any of the ϵ's is zero, that is, one of the binary variables is unbiased, the resulting probability function will be unbiased. Also, performing an XOR with a constant variable having $p_i = 0$ or $p_i = 1$ i.e a maximum bias of $\epsilon_i = \pm 1/2$ will not reduce the bias.

In our algorithm we will apply the XOR operation in a sequential manner, performing in each step the operation between a variable X_i and the result $Y = X_1 \otimes X_2 \otimes \ldots \otimes X_{i-1}$ of the repeated XOR up to that variable i. Therefore, we take a closer look on the expected result of XOR for two variables.

Lemma 3.2. *If X and Y are independent random binary variables with expectations $E(X) = \mu$ and $E(Y) = \nu$ then*

$$E(X \otimes Y) = \mu + \nu - 2\mu\nu \tag{4}$$

$$= \frac{1}{2} - 2(\mu - \frac{1}{2})(\nu - \frac{1}{2}) \tag{5}$$

Proof. Following from the logical table of the XOR, for two bits a and b, $a \otimes b$ equals 1 if $a = 0$ and $b = 1$ or if $a = 1$ and $b = 0$.

Thus, $E(X \otimes Y)$ can be written as

$$\begin{aligned} E(X \otimes Y) &= (1 - \mu)\nu + \mu(1 - \nu) \\ &= \mu + \nu - 2\mu\nu \\ &= \mu + \nu - 2\mu + \frac{1}{2} - \frac{1}{2} \\ &= \frac{1}{2} - 2\left[\mu\nu - \frac{\mu}{2} - \frac{\nu}{2} + \frac{1}{4}\right] \\ &= \frac{1}{2} - 2(\mu - \frac{1}{2})(\nu - \frac{1}{2}) \end{aligned}$$

3.1 XOR Based Multivariate Dependency Detection

The Piling-Up Lemma is successfully used in linear cryptanalysis to construct linear approximation to the action of non-linear block ciphers. In this application, the X_i-s are approximations to the substitution-boxes of block ciphers for which the biases are trivial to measure. The attack relies on performing a costly search for finding combinations of input and output values that have very high biases i.e probabilities very close of zero or one.

Similarly, we perform a search to find groups of variables for which the actual probability mass of the result obtained by performing the XOR greatly differs

from the value predicted by the Piling-Up Lemma. For these cases the high bias must come from the fact that the assumption of non-determinism is not satisfied. Thus, there is an underlaying (higher-order) statistical dependence between the inputs.

The proposed metric has a great complexity advantage, as performing k consecutive XOR operations in linear. While the approach is efficient in detecting multivariate dependences, we still have to perform an ample search to find the higher-order groups of variables that are dependent. In the next section we apply this multivariate dependence detection technique to determine all relevant parents for the Bayesian network building task.

4 Bayesian Network Building

In Bayesian network building, the goal is to decide the set $par(A_i)$ for each attribute, with the restriction, that adding the edges between and attribute and its parents must not result in a cycle.

To detect all dependencies, up to a predefined bounded size k in one step, for each attribute we compute the repeated $\otimes$ between the attribute and all possible combinations of other variables up to the threshold k. For each combination, we compute the difference between the percentage of zeros in the result as predicted by the Pilling-Up lemma and the actual outcome percentage. For each attribute, we retain the combinations that yield the biggest discrepancies.

In the network building phase, we process the attributes in a random order. For each attribute A_i, we sequentially assign the potential parent set $par_j^*(A_i)$ to be the j^{th} combination of variables with the highest bias, as quantified with the help of the Pilling-Up lemma in the previous step. In this way we process a prefixed top S_{nr} interacting subsets for each attribute. For every subset, we process each potential parent $p^*(A_i)$, $p^*(A_i) \in par_j^*(A_i)$, and if adding an edge between the attribute and its potential parent does not result in a cycle, $p^*(A_i)$ becomes a parent of A_i: $par_j(A_i) = par_j(A_i) \cup p^*(A_i)$. From all the obtained and tested parent subsets for each attribute, we choose attribute and its parents that maximizes a given discriminative scoring function, in our case the Bayesian Dirichlet metric [8].

The search stops when we determined the parents of each attribute, or when considering the extension of the network does not result in improvements. The outline of this parent search procedure is outlined in Algorithm 1.

4.1 Test Suite

To assess the performance of the proposed search method, we built some artificially generated test samples that contain various types of multivariate interactions. We consider 10 variables $X_1, \ldots, X_{10}$, sampled 5000 times.

Algorithm 1. Constructing a Bayesian network that is able to capture higher-order interactions up to a prefixed order k

```
1  BN ← EmptyNetwork();
2  foreach attribute A, iterating by i do
3      foreach e possible combinations of variables that do not contain A_i, up to
       size k, iterating by j do
4          Measure the entropy distillation bias between A_i and e_j and retain the
           S_nr combinations with highest biases in M(i,:);
5  repeat
6      for i=randompermutation(1:n) do
7          if HasParents(i) then
8              continue;
9          for j=1:S_nr do
10             par*(A_i) ← M(i,j);
11             par(A_i) ← EliminateCycles(par*(A_i));
12             BN* ← ExtendNetwork(BN, A_i, par(A_i));
13             if Score(BN*) > Score(BN) then
14                 BN ← BN*;
15 until No improvement was found;
16 return BN;
```

The *first* data set contains two highly noisy features:

- A highly noisy conditioning, where whenever three out of the four first variables are one, X_5 is also set to 1 with a probability of 0.5:

$$\text{if } (sum([X_1, X_2, X_3, X_4]) == 3) \text{ and } (rand \leq 0.5) \text{ then } X_5 = 1$$

- A noisy feature based on a parity function conditioning where if variables X_6, X_7, X_9, X_{10} have an even number of ones, X_8 is set to 0 with a 0.8 probability:

$$\text{if } (parity([X_6, X_7, X_9, X_{10}]) == true) \text{ and } (rand \leq 0.8) \text{ then } X_8 = 0$$

In the *second* dataset we reduce the amount of explicit noise but introduce an overlap between the two features, which are:

- We have the noisy conditioning, where whenever exactly half of a group of six variables are 1, X_5 is also set to 1 with a probability of 0.95:

$$\text{if } (sum([X_1, X_2, X_3, X_4, X_9, X_{10}]) == 3) \text{ and } (rand \leq 0.95) \text{ then } X_5 = 1$$

– Again a noisy feature based on a parity function conditioning:

$$\text{if } (parity([X_6, X_7, X_9, X_{10}]) == true) \text{ and } (rand \leq 0.9) \text{ then } X_8 = 0$$

In the *third* dataset we introduce an interplay between the features, where the realization of the first feature may inhibit the realization of the second one:

– We use again the first conditioning, from dataset one.
– A feature which may be short circuited by the realization of the first feature: if $X_5 == 1$, the feature regarding X_8 is not expressed.

$$\text{if } (sum([X_6, X_7, X_9, X_{10}]) == 3) \text{ and } (X_5 == 0) \text{ then } X_8 = 0$$

4.2 Results

For each test case, we generated 50 instances and tested the proposed method against the classical Bayes network model building K2 algorithm [9], which extends a current model by performing one arc operation at the time. The allowed in degree in the classic search and the k parameter in the proposed method was set to 6, thus both methods could consider up to 6 parents. The number of analyzed possible parent sets S_{nr} was set to 5.

For each batch of 50 runs, we recorded the best network found, its score, the worst and the average score. Because the data is stochastically generated and incorporates noise, the exact quantity of this values is of a little importance. The same network structure will score differently when evaluated on different noisy samples. Nevertheless, these values may be used to make qualitative assessments, in the cases where the worst result of one method surpasses the best network score or the average score found by the other method.

More important aspect regards the methods ability to extract the same structure from different samples of noisy data. We measure this robustness by comparing the best and worst scoring network out of each batch of 50 runs. If the adjacency matrix of the two networks is not similar (one can not be transformed into the other one by using only row and column swapping), implies that the search method may find different network topologies on different runs.

The numerical scores are presented in Table 1. The plot of the best networks found for each of the three cases are presented in Figures 1, 2, 3.

In the first case, where there is a high amount of noise, the classical approach can not detect the real structure, the network is filled with spurious connections where often an attribute is accounted as the parent of all other attributes following after. Observe for example in Figure 1, that Node 1 is attributed as parent for all other nodes. On the other hand, even with such a high amount of noise, the extended multi-parent search is able to detect the correct topology of the network.

For the second dataset it is expected that the classical approach is not able to detect the parity, multivariate interaction as it would need to add at least six arcs at once to reveal this interaction. Furthermore, as this feature overlaps with

Table 1. The performance of the proposed and classical methods on the three test suites. The multi-parent extended search worse results are better than the best scores obtained by the classical search method in all cases.

	Best	Worst	Average	Std.	Robust
Test suite 1					
Classic	-34128.06	-34269.11	-34204.73	32.89	No
Extended	-33662.24	-33826.25431	-33748.88	37.67	Yes
Test suite 2					
Classic	-33876.23	-34040.17	-33950.13	36.63	Yes
Extended	-33063.69	-33308.58	-33192.06	52.87	Yes
Test suite 3					
Classic	-34172.68	-34298.18	-34228.73	27.03	No
Extended	-33915.10	-34065.95	-33982.97	29.87	Yes

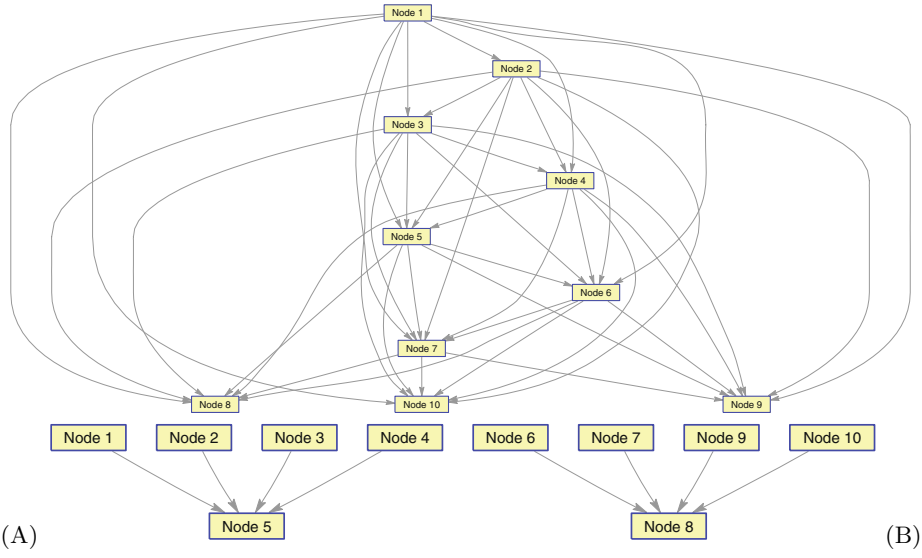

Fig. 1. Best networks found by the classical method (A) and the multi-parent extended search (B) on the first test suite

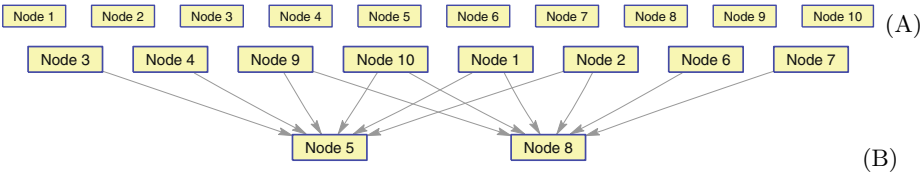

Fig. 2. Best networks found by the classical method (A) and the multi-parent extended search (B) on the second test suite

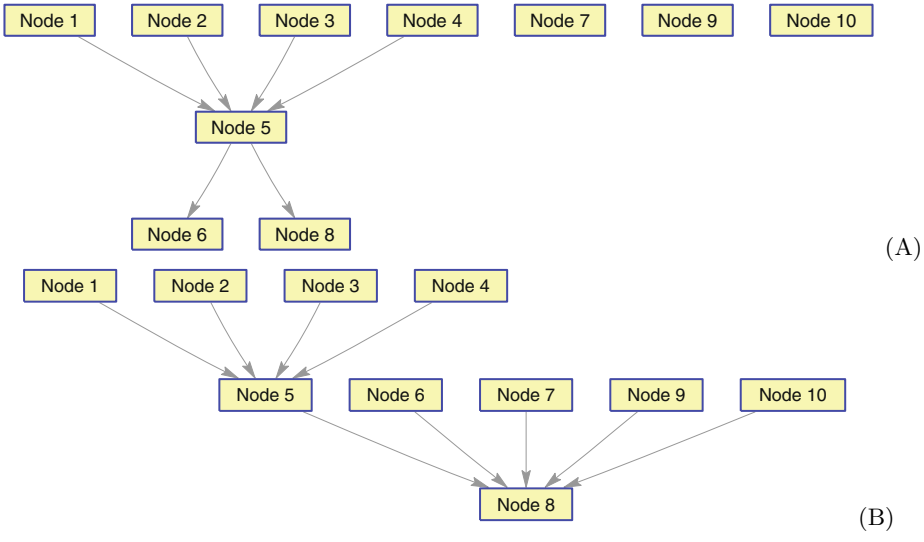

Fig. 3. Best networks found by the classical method (A) and the multi-parent extended search (B) on the third test suite

the other feature which also spans across six variables, the method is unable to account for useful relations and returns the empty network, without edges, in all cases. Please note by looking at the best and worst score in Table 1 for test suite 2, how the same empty network may score differently when presented with different test data. The proposed method is again able to find the correct structure, as we allowed the feature space exploration up to six combined variables, which is also the length of the highest multivariate relation.

On the third case, the classical method is able detect the interactions influencing attribute 5 and its relation to attribute 8, while failing to model the synergic interaction of the other variables. Sometimes, as depicted in Figure 3 A, it reports attribute 5 as linked to other variable different from attribute 8, but this result is rarely achieved. By modeling all interactions up to size six in the feature space, the multi-parent search is able to correctly decipher the interplay between the two features.

For all cases, as shown above, the extended search found qualitatively better networks; the worst scoring results of the proposed method were always better than the best results returned by the classical method. As it does not contain stochastic components, the proposed showed robustness, finding the same topology on different runs.

5 Conclusions

Usual Bayesian network building starts by exploiting pairwise dependencies. When no such relations are available a successful approach must do k-wise multivariate interaction search.

In this paper a search algorithm for constructing Bayesian networks from binary data was developed, where all dependencies of each attribute is detected in one step. The proposed method has demonstrated a great ability to identify simpler and synergic multivariate interactions even in the case of noisy feature interplay, where considering one edge addition at the time is fruitless.

While it uses a small number of model evaluations and it is much more effective than doing greedy search using a k-wise stochastic edge search operator, the extended multi-parent search is still very costly in terms of building and evaluating all combinations of variables, having an $O(n^k)$ complexity. Fortunately, backtracking algorithms are very easy to parallelize as processing different paths in the search tree is an embarrassingly parallel task, with no communication overhead [10,11]. Parallel backtracking scales very well with the number of available processors. Therefore, future effort will focus on parallelizing the higher-order dependency detection search.

Acknowledgments. This research is supported by the Sectoral Operational Program for Human Resources Development 2007-2013, co-financed by the European Social Fund, within the project POSDRU 89/1.5/S/60189 with the title "Postdoctoral Programs for Sustainable Development in a Knowledge Based Society". We also acknowledge the financial support of the Sapientia Institute for Research Programs (KPI).

References

1. Bouckaert, R.: Properties of Bayesian belief network learning algorithms. In: Uncertainty in Artificial Intelligence: Proceedings of the Tenth Conference, July 29-31, p. 102. Morgan Kaufmann (1994)
2. Chickering, D., Geiger, D., Heckerman, D.: Learning Bayesian networks is NP-hard. Technical Report MSR-TR-94-17, Microsoft Research (November 1994)
3. Koivisto, M., Sood, K.: Exact bayesian structure discovery in bayesian networks. J. Mach. Learn. Res. 5, 549–573 (2004)
4. Rissanen, J.: Modelling by the shortest data description. Automatica 14, 465–471 (1978)
5. Pelikan, M.: Hierarchical Bayesian optimization algorithm: Toward a new generation of evolutionary algorithms. Springer (2005)
6. Cover, T.M., Thomas, J.A.: Elements of information theory. Wiley-Interscience, New York (1991)
7. Matsui, M.: Linear Cryptanalysis Method for DES Cipher. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 386–397. Springer, Heidelberg (1994)
8. Heckerman, D., Geiger, D., Chickering, D.: Learning Bayesian networks: The combination of knowledge and statistical data. Machine Learning 20(3), 197–243 (1995)
9. Cooper, G.F., Herskovits, E.: A bayesian method for the induction of probabilistic networks from data. Mach. Learn. 9(4), 309–347 (1992)
10. Kouril, M., Paul, J.L.: A parallel backtracking framework (bkfr) for single and multiple clusters. In: Proceedings of the 1st Conference on Computing Frontiers, CF 2004, pp. 302–312. ACM, New York (2004)
11. Herley, K.T., Pietracaprina, A., Pucci, G.: Deterministic parallel backtrack search. Theor. Comput. Sci. 270, 309–324 (2002)