CCGA-BN Constructor: A Bayesian Network Learning Approach

Maryam Feroze
Department of Computer Science/UBIT, University of Karachi, Pakistan

Muhammad Saeed
Department of Computer Science/UBIT, University of Karachi, Pakistan

Nasir Touheed, Ph.D.
Faculty of Computer Science, Institute of Business Administration, Karachi, Pakistan

ABSTRACT
This paper presents a tool CCGA-BN Constructor for learning Bayesian network that uses cooperative co-evolutionary genetic algorithm to learn Bayesian network structure from data. The problem has been broken down into two sub-problems: (a) to find the optimal nodes'ordering and (b) to find the optimal adjacency matrix of the graph. Both the sub-problems' solutions are then combined to produce the optimal structure. CCGA-BN constructor used Bayesian score for networks having nodes with more than two states and BIC for network having bistate nodes. The findings of this paper are compared against the original structures and the results show a lot of promise.

Keywords
Bayesian network, cooperative co-evolutionary genetic algorithm, structure learning, Bayesian score, BIC.

1. INTRODUCTION
Bayesian Network(BN) is a model, which represents and reasons under uncertainty, represented as Directed Acyclic Graph (DAG) whose vertices signifies the domain variables and the edges, if present, establish the conditional dependence between variables.

If defineformallya Bayesian Network ***N*** is consists ofa DAG G = (V, E) with verticesVand edgesE, a set of discrete random variables, X = {X_1, . . . , X_n}, represented by the vertices of G and a set of conditional probability distributions(CPDs), P, containing a distribution, $\boldsymbol{P(X_i|pa(X_i))}$, for each $X_i \in X$. A Bayesian network decomposes the joint probability distribution $\boldsymbol{p(X) = p(X_1,\ldots X_n)}$ into a product of conditional probability distributions over each variable given its parents:

$$p(X) = \prod_{i=1}^{n} p(X_i \mid pa(X_i))$$

where$pa(X_i)$ be the set of parents of Xi in G.

A Bayesian network can be constructed manually with the help of services, skills and knowledge of the domain expert or can be learned from the observed data.The DAG of a BN was constructed by the collaboration of domain experts until the early 1990s and then the conditional probabilities of the DAG were either calculated from the available data or by the wisdom and knowledge of experts were estimated, also the combination of both approaches was practiced. Construction of a BN manually by experts can be a laborious and can take long time especially in the case of large networks. Thus, the researchers proposed methods that enable to learn BN from the empirical data set. Learning BN from data is composed of two tasks (i) Learning structure of BN that is DAG (ii) Learning BN parameters that is CPDs for each variable. In this paper a method for learning structure of the BN from the data is used and parameters are then learned in conventional way.

The task of learning DAG of a BN from a data set can be formulated as an optimization problem[1][2] that has been proven to be NP-Hard[3] as learning the DAG from data by exhaustively considering all possibilities is not feasible, regardless of the size of the data, since the possible DAGs that can be constructed grows super-exponentially in the number of vertices.Hence, this either requires search algorithms that uses sub-optimal heuristics or algorithms that produces optimal solution under certain assumptions.

Coevolution is said to be a mutual evolutionary variation between species that interact with each other. Coevolution, based on the method of calculation of the fitness of individuals, can be based either on competition or on cooperation. When the coevolution is competitive, the fitness is computed by direct competition of different species whereas in the cooperation, the fitness of an individual species depends on the collaboration with other species. Cooperative Coevolutionary Genetic Algorithm (CCGA) can be applied to the problems that can be decomposed into number of independent parameters/species that coordinates with each other in order to compute the fitness of the complete individual. Fitness to an individual of a particular species can be assigned by merging it with the selected individuals of the other species to form one complete individual whose fitness can be computed in the normal fashion, which is then used to assign fitness to the individual component. The task of structure learning can be divided into two optimal sub-problems (a) to find an ordering of the nodes that is optimal and (b) to find an optimal connectivity matrix. Thus, the problem can be modeled and solved using CCGA which uses two species one for each subproblem.

Previously, in this field of researchdifferent evolutionary algorithms such as Genetic Algorithms (GA) and its variants have been used [4][5][6][7], all these approaches address this problem as one complete problem of optimal DAG generation from data. Cooperative Coevolutionary Genetic Algorithm is also used in [8] for learning the structure of BN from datasets. In this paper, the problem is representedin the similar way as in [8] but algorithm is a variation of conventional CCGA.

Besides this section, this paper is organized as follows. In the following section, an introduction to Bayesian networks learning, structure learning problem and scoring methods is discussed. In Section 3,we present overview CCGA-BN andin section 4the results obtained for some networks and the comparison of obtained structures with the original ones are discussed. We show that the approach of BN-CCGA performs satisfactory. Finally, we conclude in last section .

2. BAYESIAN NETWORK LEARNING

2.1 Parameter Learning

BN parameters are estimated form the data set in the way that satisfies the following theorem:

Theorem: *Suppose we specify a Bayesian network for parameter learning in the case of binomial variables X_i and assign for all iand j*

$$a_{ij} = b_{ij} = N / 2q_i$$

whereN is a positive integer and $q_i$is the number of instantiations of the parents of the ithvariable. a_{ij} and b_{ij} specifies the two states of the variable X_i and for jth state of the $pa(X_i)$. Then the resultant Bayesian network has equivalent sample size N, and the joint probability distribution in the Bayesian network is uniform.

The prior probability of the variables is calculated as:

$$P(X_i = a_{ij}) = a_{ij} / (a_{ij} + b_{ij})$$

$$P(X_i = b_{ij}) = b_{ij} / (aij + b_{ij})$$

The updated probability calculated from the data set of M elements is:

$$P(X_i = a_{ij}) = (a_{ij} + s_{ij}) / (N + M)$$

$$P(X_i = b_{ij}) = (b_{ij} + t_{ij}) / (N + M)$$

Pair (s_{ij}, t_{ij}) is used to represent the counts for the ***ith*** variable's instances in the data when the variable's parents have their ***jth*** value.

For Multinomial Variables X_i, having ***s*** states, for its ***rth*** state a_r we will have:

$$a_{rij} = N / sq_i$$

The prior probability of X_i is:

$$P(X_i = a_{rij}) = a_{rij} / \sum_{r=1}^{s} a_{rij}$$

2.2 Structure Learning

The problem of structure learning is to learn a DAG from data that fulfills the Markov condition with the probability distribution P that is generating the data, where P is not known. Thispracticeof learning such a DAG is called model selection.

Bayesian Network N with DAG G and Probability distribution P satisfies the Markov condition if and only if P is equal to the product of its conditional distributions of all nodes given their parents in G, whenever these conditional distributions exist.

Approaches to learning the structure of a Bayesian network have been classified into three classes [9]. The *constrained-based* approach derives set of (in)dependencies from the data that are used in learning structure as constraints [10]. The *score-based* approach calculates a scoring function for the structures that are learned from the data, the scoring function is the measure of the fitness of structure on the observed data.Lastly, the *Bayesian model averaging* is a collaborative approach that generates structure from the integration of different structures and inference is performed by averaging the results provided from different models learned. In this paper, the focus is on the approach of score-based learning.

There are two common approaches in score-based approach of structure learning, one is the Bayesian scoring and the other one is Bayesian Information Criterion Score. In this paper both of the scoring approaches are used, for network with only bi-state variables BIC is used to learn the best DAG G, and for data having nodes with more than two states Bayesian Scoring is used.

2.3 Score based Learning

In score-based structure learning, scoring function Φ is considered to quantify how well the DAG fits the data. Our problem can be defined as*"Given a data D = {y_1, . . . ,y_N} and a scoring function Φ, we need to find a DAG G that maximizes the value Φ(G, D)".*

A Bayesian score function **Φ**, isvalued by evaluation of the posterior probability of a graph G given the data *D*:

$$P(G|D) = \frac{P(D|G)P(G)}{P(D)}$$

$$P(G|D) \propto P(D|G)P(G)$$

where the equality is given by using Bayes' theorem. The denominator in above equation doesn't help in distinguishing between the different structures and can be disregarded as it is simply a normalizing factor. Similarly P(G) of each possible G for the data is same, thus P(D | G) is only what we need to compute for scoring.

For variables X_i[2] proposed the use of a Dirichlet parameter prior for all parameters in the network, then $P(D|G)$ can be obtained as:

$$P(D|G) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(N'_{ij})}{\Gamma(N'_{ij} + N_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(N'_{ijk} + N_{ijk})}{\Gamma(N'_{ijk})}$$

where n represents the count of variables in the network, $r_i$is the count of all possible values for the variable X_i, $q_i$is the total number of possible joint assignment of values to the parents of X_i, $N_{ijk}$is the number of occurrences of configurations of variables and their parents, $N'_{ijk}$are the prior counts of occurrences of variables and their parents calculated as $N_{ij}=\sum_{k=1}^{r_i} N_{ijk}$ and $N'_{ij}=\sum_{k=1}^{r_i} N'_{ijk}$, and Γ is the Gamma function, which satisfies $\Gamma(m) = (m-1)!$. The logarithm of equation is more practical to use since it is more manageable to compute numerically.

The total number of possible structures *r(n)*for a network with *n* nodes, can be found by the recursive formula[11]:

$$r(n) = \sum_{k=1}^{n} (-1)^{k+1} \binom{n}{k} 2^{k(n-k)} r(n-k)$$

*i.e., r(n),*which is the function based on the number of nodes, is super-exponential in *n*. When $n = 6$, $r(n) = 3, 781, 503$, i.e.this many possible DAGs. This result reaches approximately 4.2×1018 when $n = 10$.

2.4 Bayesian Information Criterion

Another common scoring criterion for BN structure learning is Bayesian information criterion (BIC) scores, which is as follows:

$$BIC(G:D) = \ln\left(P(D|\hat{P}, G)\right) - \frac{d}{2} \ln m$$

where m is the number of data items, d is the number of parametersof the DAG model, and $\hat{P}$is the set of maximum likelihood values of the parameters.

3. BN-CCGA CONSTRUCTOR

The proposed solution to the Bayesian network structure learning problem from a fully observable data set is established on the cooperative coevolutionary genetic algorithm (CCGA) proposed by Potter and De Jong [12]. In this solution onecomplete problem is divided into two dependent subproblems(a) to find an optimal ordering of the nodes and (b) to find an optimal connectivity matrix.

Therefore, two independent species,each representing a different but related sub-problem, are required that can collaborate and cooperate with each other to form a complete solution, one species to represent each subproblem. The algorithm, of CCGA-BN is given in figure

1. Initialize n individuals of Species 1
2. Initialize n individuals of Species 2
3. For m generations performs the following steps:
   - 3.1. For both species perform following steps to produce n individuals
     - 3.1.1. Select two individuals/parents P1, P2
     - 3.1.2. P1 and P2 undergoes Crossover to produce C1 and C2
     - 3.1.3. Mutate C1 and C2
     - 3.1.4. C1 from both species are united to produce one complete solution individual I1 and C2 from both species are combined to produce complete solution individual I2
     - 3.1.5. Evaluate I1 and I2
   - 3.2. New generation of both species are built by replacing the individuals of previous generation with the newly created fittest individuals.
4. Best Individual of the last generation gives the solution

Each individual in species 1, which can also be named as permutation population, is represented by ordering of the random variables $X_1, X_2, \ldots, X_n$, where parent nodes of the node present at position *i* come at any position between 1 and i-1. A fully connected Bayesian network with *n* nodes has $(n-1)n/2$ edges. An individual from the binary subpopulation is represented string $b_{1,2}, b_{1,3}, \ldots, b_{1,n}, b_{2,3}, \ldots, b_{2,n}, \ldots, b_{n-1,n}$; where $b_{i,j}$ is 1 if the node at position *i* is a parent of the node at position j, and is 0 otherwise. The length of the string is $(n-1)n/2$. Also it can be visualized as for $n = 4$ in the table1 below.

Table 1: Species 2 Individual Representation

	1	2	3	4
1	0	$b_{1,2}$	$b_{1,3}$	$b_{1,4}$
2	0	0	$b_{2,3}$	$b_{2,4}$
3	0	0	0	$b_{3,4}$
4	0	0	0	0

One complete solution i.e. structure of BN is formed by integrating one individual from both the populations in such a way that for each element at position *i* in the permutation population individualrepresents a node that is parent of the node appearing at position $i+x$(x <= number of node) if there is $c_{i,i+x}=1$ in binary population individual. For example, if permutation population individual is D, B, A, C and binary population individual is 1, 0, 1, 1, 0, 1 then the solution is the DAG with an edge from D to B and C, from B to A and from A to C.

The solution obtained will always be legal and will never have cycles because node ordering is implicitly specified and only the strictly upper triangular connectivity matrix is used to represent connectivity. Also, the solution is complete because every possible structure of Bayesian network can be represented.

Remaining details of the algorithm are summarizedin table 2given below.

Table 2: CCGA Details

Fitness Function	**Φ**for muti-valued variables BIC for bi-valued variables
SPECIES 1—Nodes Ordering	
Representation	Permutation Population representing ordering of the nodes
Initialization	Random initialization
Selection	Tournament Selection
Crossover	Cyclic Crossover with probability p_c
Mutation	Swap mutation with probability p_{mp}
Replacement	Elicit replacement
SPECIES 2 – Connectivity Matrix	
Representation	Binary Population representing connectivity matrix
Initialization	Semi-random initialization
Selection	Tournament Selection
Crossover	Two-point crossover with probability p_c
Mutation	Bit-flip with probability p_{mb}
Replacement	Elicit replacement

By using the above mentioned approach CCGA-BN Constructor is built with the parameters of algorithms discussed in the later section. CCGA-BN Constructor is implemented on language C#. Also, Smile API is used to produce output network in any of the following formats: xdsl, dsl (genie files), dne (Netica File), net (Hugin File), erg (Ergo File), dsc (Interchange). CCGA BNC reads data file from either the text document or from any of the two formats of excel files and then uses this dataset to perform the task of BN learning by using Bayesian score for multi-state variable network and uses BIC for networks with variables having only two states.

4. EXPERIMENTS

The experiment is performed on multiple networks and here analysis is presented on two of those networks. One of the two networks is composed of 9 nodes and the other one is a well-known Alarm Network.

Parameters of used in CCGA-BN Constructor are in table3.

Table 3: Parameters of CCGA

Parameter	Value
Total generations	100
Population size	100
P_{mb}	1/[n(n-1)/2]
P_{mp}	0.5
P_c	0.6

Where *n* is the total number of nodes.

For both Bayesian networks, first the data sets containing 1000 instances are generated and then CCGA-BN Constructor for learning structure is executed. Network 1 for which CCGA-BN Constructor was executed is given below:

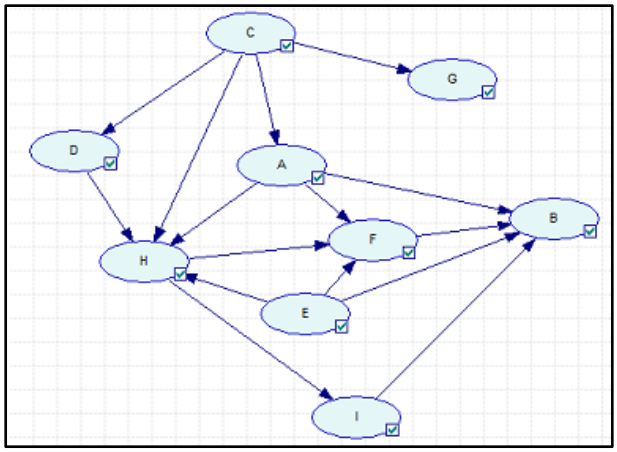

Figure 1: Network 1

The structure learned from CCGA for Network 1is shown in Figure 3.

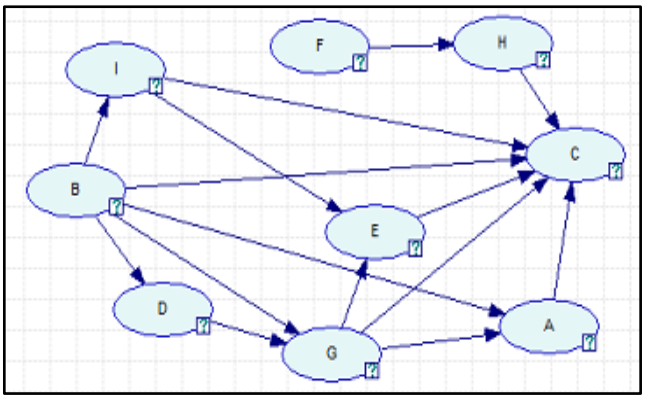

Figure 2: Resultant Network

For this network, the resultant network and the original one don't have any common edge, no converging relationship is maintained i.e. the resultant network doesn't seems to be a good output if sustaining of connections and relationships is to be considered. But if the two networks are compared on the basis of their marginal probabilities and on the inferences when 1, 2 and 3 evidences are set on some random nodes, the result is quite satisfactory. That is, the differences in the marginal probabilities are minor for most of the nodes. The comparison of these probabilities of network 1 is given in the following tables.

Table 4: Comparison of the network 1 with the resultant learned network from the CCGA-BN Constructor on the basis of Marginal Probabilities

		Marginal Probabilities (in %)		
Node	State	Original Network	Learned Network	Difference
A	TRUE	81	80	1
	FALSE	19	20	
B	Class1	22	24	2
	Class2	78	76	
C	Yes	43	46	3
	No	57	54	
D	State2	67	62	5
	State3	33	38	
E	Right	46	47	1
	Wrong	54	53	
F	Temp	57	57	0
	Perm	43	43	
G	State2	51	52	1

	State3	49	48	
H	State2	52	53	1
	State3	48	47	
I	State2	47	48	1
	State3	53	52	
Average Difference = 1.666667 Variance Difference = 2.25				

Table 4 represents the comparison between the sample network given in figure 1 and the network that is learned by CCGA BN-Constructor given in figure 3. Comparison is performed on the basis of marginal probabilities at each state of each node and as can be visible by the table the differences between the given and learned network are low on average the difference is 1.67 and on average the standard deviation is 1.5 for the difference between the probabilities. (Probabilities are given in percent)

Table 5: Comparison of the network 1 with the resultant network from the CCGA-BN Constructor on the basis of Marginal Probabilities when evidence is placed on I

		Marginal Probabilities (in %)		
Node	State	Original Network	Learned Network	Difference
A	TRUE	81	79	2
	FALSE	19	21	
B	Class1	24	28	4
	Class2	76	72	
C	Yes	43	46	3
	No	57	54	
D	State2	67	62	5
	State3	33	38	
E	Right	46	47	1
	Wrong	54	53	
F	Temp	57	57	0
	Perm	43	43	
G	State2	51	51	0
	State3	49	49	
H	State2	53	53	0
	State3	47	47	
I	Evidence on I at state 2			
Average Difference = 1.875 Variance Difference = 3.839286				

Table 6: Comparison of the network 1 with the resultant learned network from the CCGA-BN Constructor on the basis of Marginal Probabilities when evidence is placed on I and B

		Marginal Probabilities(in %)		
Node	State	Original Network	Learned Network	Difference
A	TRUE	73	69	4
	FALSE	27	31	
B	Evidence on B at state class1			
C	Yes	44	40	4
	No	56	60	
D	State2	67	61	6
	State3	33	39	
E	Right	53	47	6
	Wrong	47	53	
F	Temp	59	57	2
	Perm	41	43	
G	State2	51	52	1
	State3	49	48	
H	State2	52	53	1
	State3	48	47	
I	Evidence on I at state 2			
Average Difference = 3.428571 Variance Difference = 4.619048				

Table 7: Comparison of the network1 with the learned network from the CCGA-BN Constructor on the basis of Marginal Probabilities when evidence is placed on I and B

		Marginal Probabilities(in %)		
Node	State	Original Network	Learned Network	Difference

A	TRUE	72	69	3
	FALSE	28	31	
B	Evidence on B at state class1			
C	Evidence on C at state Yes			
D	State2	55	61	6
	State3	45	39	
E	Right	52	48	4
	Wrong	48	52	
F	Temp	59	56	3
	Perm	41	44	
G	Class2	55	55	0
	Class3	45	45	
H	Class2	52	55	3
	Class3	48	45	
I	Evidence on I at state 2			
Average Difference = 3.166667 Variance Difference = 3.766667				

The learned structure for the alarm network preserves 5 out of 47 parent-child relationships of the original network. The marginal probabilities,(in percent) calculated from the CPDs that are learned from data for the network produced by CCGA-BN Constructor, have low difference when compared with that of original network. The average standard deviation of the differences of the marginal probabilitiesof two networks is 3.56231, which reflects that the resultant network learned performs almost similar to the original network when performing inference. When evidence is placed at state *normal* of node *ArtCO2* the standard deviation of difference between marginal probabilities of learned and original alarm network is 2.773865. When evidence is placed at state *high* of node *PAP* the standard deviation is 5.061452, and when evidence is at *normal* state of *SaO2*the differences are still found to be low which is reflected by the standard deviation which is found to be 4.349598.

Time complexity is a challenge in solving the problem of learning network form data, the average time taken to by CCGA-BN Constructor for learning network from the data of network 1 is almost 15 minutes and for alarm network it was almost 5 hours when executed for dataset of 1000 records

5. CONCLUSION

We implemented CCGA-BN Constructor for achieving the task of Bayesian network learning from data sets. The approach implemented in this paper addressed the problem by decomposingit into two dependent subtasks that is to find an optimal nodes' ordering and an optimal connectivity matrix. We analyzed the performance of the algorithm on the basis of the differences in marginal probabilities between those that are learned from the data for the learned structure and those that are observed from the original network. The networks we analyzed in the paper are alarm network and another network of 9 nodes. The results showed that the solutions' obtained marginal probabilities (in percent) when compared to the original networks have low differences, on average difference is between 1 and 10. For the analysis data sets of 1000 instances are used for both the networks. Even after placing evidence on different nodes the inferred marginal probabilities have low variance/standard deviation and low average difference. But, time complexity of the solution is high and can be improved as it is a problem that is suitable for parallel computing environment as there are two independent species' populations that can be constructed side by side.

6. REFERENCES

[1] D. Heckerman, D. Geiger, and D. M. Chickering. "*Learning Bayesian networks: The combination ofknowledge and statistical data"*. Machine Learning, 20:197–243, 1995.

[2] G. F. Cooper and E. Herskovits.*"A Bayesian method for the induction of probabilistic networks from data".Machine Learning*, 9:309–347, 1992.

[3] D. M. Chickering, D. Geiger, and D. Heckerman. "*Learning Bayesian Networks is NP-Hard".* Technical report, Microsoft Research, 1994.

[4] C. Cotta and J. Muruzabal.*"On the learning of Bayesian network graph structures via evolutionary programming"*. In Proceedings of the 2nd European Workshop on Probabilistic Graphical Models, pages 65–72, 2004.

[5] P. Larranaga and M. Poza.*"Structure learning of bayesian networks by genetic algorithms: A performance analysis of control parameters"*. IEEE Journal on Pattern Analysis and Machine Intelligence, 18(9):912–926, 1996

[6] M. Wong, W. Lam, and K. Leung.*"Using evolutionary programming and minimum description length principle for data mining of Bayesian networks"*. IEEE Transactions PAMI, 21(2):174–178, 1999.

[7] M. L. Wong, S. Y. Lee, and K.-S.Leung.*"A hybrid data mining approach to discover bayesian networks using evolutionary programming"*. In GECCO, pages 214–222, 2002.

[8] Arthur Carvalho, David Cheriton. *"A Cooperative Coevolutionary Genetic Algorithm for Learning Bayesian Network Structures".*School of Computer Science, University of Waterloo, Ontario, Canada. July 2011.

[9] D. Koller and N. Friedman. "*Probabilistic Graphical Models: Principles and Techniques"*. MIT Press, 2009.

[10] Spirtes, Glymour, AndScheines. *"Causation, Prediction And Search"*. Springer-Verlag, 1993.

[11] R. W. Robinson. "*Counting Unlabeled Acyclic Digraphs".* In Combinatorial Mathematics V, volume 622 of Lecture Notes in Mathematics, pages 28–43,1977.

[12] M. Potter and K. De Jong."*A cooperative coevolutionary approach to function optimization"*. In Third Conference on Parallel Problem Solving from Nature, pages 249–257, 1994.

[13] A. Delaplace, T. Brouard, and H. Cardot."*Two Evolutionary Methods for Learning Bayesian Network Structure"s*. In *Computational Intelligence and*

Security, pages 288–297. 2007.

[14] Mitchell A. Potter and Kenneth A. De Jong *"A Cooperative Coevolutionary Approach to Function Optimization"*.Computer Science Department, George Mason University, Fairfax, VA 22030, USA.

[15] J. Pearl. *"Probabilistic reasoning in intelligent systems: networks of plausible inference"*. Morgan Kaufmann, 1997.

[16] Alexandra M. Carvalho *"Scoring functions for learning Bayesian networks"*.Inesc-id Tec. Rep, 2009