Bayesian Network Building for Diagnosis in Industrial Domain based on Expert Knowledge and Unitary Traceability Data

Thierno M. L. Diallo, Sébastien Henry, Yacine Ouzrout

Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC-PapersOnLine 48-3 (2015) 2411–2416

Bayesian Network Building for Diagnosis in Industrial Domain based on Expert Knowledge and Unitary Traceability Data

Thierno M.L. DIALLO*. Sébastien HENRY* Yacine OUZROUT**

**DISP laboratory, University of Lyon, University Lyon 1, France*
{Thierno.Diallo, Sebastien.Henry}@univ-lyon1.fr
***DISP laboratory, University of Lyon, University Lyon 2, France*
Yacine.Ouzrout@univ-lyon2.fr

Abstract: This paper presents the CBNB (Causal Bayesian Networks Building) algorithm for the causal Bayesian Network construction. This algorithm is designed for diagnosis models in the industrial domain. It uses expert knowledge and operates process and product traceability data. The first phase of this algorithm consists of exploiting expert knowledge and properties of the application domain for allocating the variables at different levels of causality. This phase results in a cascade arrangement of the system's variables starting with the root causes and ending with the ultimate effects passing through one or more intermediate levels. In the second phase based on the unitary traceability data, the CBNB algorithm then allows to determine the causal relationships existing between variables. We provide the necessary assumptions and the theoretical justifications for the proposed algorithm. We have conducted empirical studies assessing the ability of the algorithm to provide the true network from synthetic data on three benchmarks whose nodes are arranged in cascade. The results of comparative analysis have shown that the CBNB algorithm outperforms GS and MMHC, two state-of-the art structural learning algorithms in terms of ability to rebuild the true network.

Keywords: Bayesian Network, Structural Learning Algorithm, Industrial Diagnosis, Big Data.

1. INTRODUCTION

With the continuing growth of the industrial processes instrumentation and the development of automatic data collection tools and technologies (RFID, Data Matrix, etc.), companies collect more and more data. The potential of this data for industrial performances improvement is now obvious. We consider in our work traceability data, especially unitary traceability data (including process and product data). Generally, the traceability unit is an aggregation of several items (e.g. a lot or a pallet). This level of detail is enough to have an accurate picture of the conditions of production for batch production but not for job production and flow production. The unitary traceability enables a serialized unique identification at the item level and allows to know accurately the process parameters values of each item. The process parameters to trace are raw materials and ingredients making up the product, the transformation processes and distribution historical and location of the product after delivery. In addition to these process data, the product features are recorded. By observing industrial practices, we found that traceability performed internally by different departments (manufacturing, maintenance, etc.) are managed separately and reconciliation are not made between them. Another observation we made is that there is no general rule on what data to collect especially for internal traceability. External traceability is less challenging to manage and the EPC Global standards provide a data model for traceability along a supply chain. In previous work (Diallo et al., 2014), we have developed a data model for internal unitary traceability. The data is aggregated by production order. The production processes are divided into segments and for each segment, the data related to process and product are collected. This proposed data model allows to know for each item, the process parameters of its manufacture from historical time series data sets. This data can be used for diagnosis function development, supply chain optimization, life cycle management, etc. However, the development of methods and models to process this data is a new challenge for academia and industry nowadays. Bayesian Networks (BN) are one of the various analytical tools that have been used to derive knowledge from data. The main advantage of BN over other artificial intelligence tools is that they allow to combine, on the one hand, certain and uncertain knowledge and on the other hand, they allow to exploit both data and expertise. To set up a Bayesian model, two elements have to be defined: the structure of the network (nodes and arcs) and the network parameters (conditional probabilities tables). The determination of the structure is by far the most challenging (Cheng et al., 2002) (Ramirez and Piqueras, 2006). In general, there are three approaches for learning BN from data: constraint-based approach, search-and-score approach and hybrid approach. The constraint-based approach uses the conditional dependence or independence relationships between variables derived from the data to guide the network construction. The second approach consists of, among all possible networks, identifying the one that maximizes a score measuring the adequacy of the network to the data. The hybrid approach combines the principles of the two previous. In our literature review, we found that the Bayesian network building still remains a lock. Indeed, existing construction algorithms

10.1016/j.ifacol.2015.06.449

come up against the explosion in the number of variables. Learning BN from data is an NP-hard problem (Tsamardinos et al., 2006). The most current heuristics are not efficient on high dimensional data with a limited sample size. Application examples of BN published in the literature generally focus on a dozen variables. The issue related to the construction of the network for industrial applications in the case of a high number of variables is rarely addressed. However, industrial applications involve, in some cases, hundreds of parameters with millions of possible records. Algorithms whose complexity is exponential relative to the number of variable are therefore not applicable to such problems. In this paper, we propose a structural building algorithm applied to industrial processes. We prefer the word "building" to the word "learning" in the name of our algorithm because it strongly incorporates expert knowledge. The learning phase of the algorithm receives as input the variables divided into different levels of causality defined by domain experts. Based on unitary traceability data, the algorithm then determines the causal relationships existing between variables.

The remainder of this paper is organized as follows: concepts and theoretical results necessary for the understanding of this paper are presented in Section 2. Our algorithm is described in Section 3 followed by the presentation of settings and results of the experimental studies in Section 4. Finally, Section 5 summarizes our contributions and announces perspectives of this work.

2. PRELIMINARIES

A Bayesian Network (BN) is a DAG (Directed Acyclic Graph) G represented by the couple (V, E), where V is a set of vertices encoding joint probability distribution and E is a set of oriented edges or arcs linking the vertices. The vertices correspond to "random" variables (discrete or continuous) and the edges represent the relationship (possibly causal) amount the variables. A marginal or conditional probability distribution table is associated to each vertex. The set of probabilities of the network is denoted by P.

Terminology. If two nodes are connected by an arc, the node at the origin of the arc is said *parent* of the node at the extremity of the arc. Conversely, the node at the end of the arc is said *child* of the node at the origin. If the two nodes are connected by more than one arc (a chain), the node at the origin of the chain is said *ancestor* of the node at the end of the chain. The latter is said *descendant* of the node at the origin of the chain. A node without a parent is called a root node and a node without son is said *leaf* node. Any node which is neither root nor leaf is called *intermediate* node (Korb and Nicholson, 2003).

The couple (G, P), with G = (V, E) a DAG, is a Bayesian network if it satisfies the Markov condition.

The Markov condition. A directed acyclic graph G over V and a probability distribution P(V) satisfy the Markov condition if and only if for every W in V, W is independent of V\(Descendants(W) ∪ Parents(W)) given Parents(W) (Spirtes et al., 2000).

The Markov condition establishes a set of independence relations on the DAG G. Indeed, the Markov condition ensures that all direct dependencies in the system being modelled are explicitly shown via arcs (Korb and Nicholson, 2003). Whether two nodes are not connected by an arc, then they are conditionally independent.

D-separation. Given a DAG G and X, Y, two vertices of G with $X \neq Y$. Let W be a set of vertices of G not containing X and Y. We say that X and Y are d-separated given W if and only if there are not undirected path U between X and Y such that (i) each collider on U has a descendant in W and (ii) no other vertex of U is in W.

We say that X and Y are d-connected given W if and only if they are not d-separated with respect to W (Spirtes et al., 2000).

3. THE CAUSAL BAYESIAN NETWORKS BUILDING (CBNB) ALGORITHM

CBNB is a mixed algorithm combining knowledge and data. It is applicable to causal models construction such as industrial diagnosis systems. The CBNB algorithm has two phases: the allocation phase and the causal relationships learning phase. Expert knowledge is first used to allocate the system's variables to the predefined levels of causality and finally the data is employed to determine causal relationships between the system variables.

3.1 CBNB algorithm: the allocation of the system's variables

The existing algorithms in the literature incorporate little or no expert knowledge. However in some domains, these knowledge are very valuable to be ignored. This is especially the case of domains with huge number of variables, knowing that most of these algorithms are exponential relative to number of variables. Examples of information that we can obtain from domain experts are (Cheng et al., 2002) (Riascos et al., 2007): identification of root nodes, identification of leaf nodes, existence (or absence) of a relationship between two nodes and definition of an (partial or complete) order on the variables.

In the CBNB algorithm, the variables are assigned to different levels of causality. These levels should then be ordered from the root level (consisting of root nodes) to leaf level (composed of leaf nodes). There can be one or more intermediate levels between these two extreme levels. A variable belonging to a given level is likely to directly influence one or more variables belonging to the consecutive lower level. Direct dependency relationships are not possible between nodes of non-consecutive levels. Levels must be homogeneous in terms of causality such that a variable cannot influence a variable with which it shares the same level of causality. We designate this particular network configuration by the cascade arrangement (c.f. Figure 1)

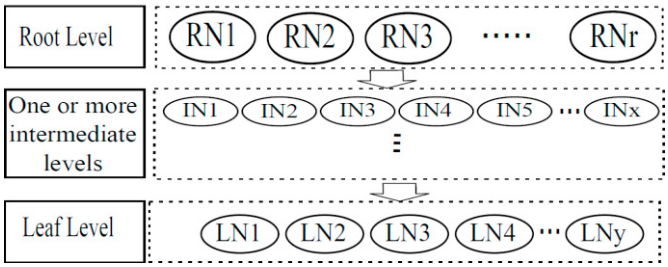

Figure 1. The cascade arrangement

For industrial diagnosis models, the allocation of variables to homogeneous causal levels and the arrangement of these levels can be realized in an intuitive way. For these models, the following distribution could be considered: the root level might consists of machine and process parameters, then the functioning modes and defects types could form the intermediate levels and finally the symptoms could be considered as leaf nodes.

Causal Bayesian networks applied to industrial diagnosis that we have consulted in the literature have for the most this cascade arrangement. Examples of these networks can be found in (Ramirez and Piqueras, 2006), (Chen et al., 2012), (Weidl et al., 2005), (Dey and Stori, 2005) and (Przytula and Thompson, 2000). These examples and other examples of applications of Bayesian networks that can be found in the literature are generally limited to a few tens nodes. These diagnosis models address most of the time a single piece of equipment. The diagnosis approach we are interested in is to analyse all industrial processes to identify causes of product defects or performances degradation. In such cases (especially for large-scale and complex processes), the number of parameters to be taken into account is in the hundreds. Hence the need for an adapted algorithm since the existing algorithms are not efficient on high dimensional data with a limited sample size. We propose an algorithm more effective by incorporating expert knowledge which have led to the cascade arrangement.

3.2 CBNB algorithm: the causal relationships learning

3.2.1. The theoretical foundations

Assumptions:

A1: The causal model to construct is a Bayesian network

A2: This causal Bayesian network is arranged in cascade as described in section 3.1

The first assumption implies that the graph G (V, E) to be achieved is a DAG and the couple (G, P), where P is the set of the probabilities of the network, fulfils the Markov condition.

The Markov condition ensures that any conditional independence entailed by the graph G is also present in the probability distribution P. In other words, if there is no relationship between two nodes (linked by arc or chain), then these two nodes are conditionally independent. But the reverse is not always true. That is to say, when two nodes are conditionally independent, it does not always means an absence of an oriented path towards one another (see example in (Spirtes et al., 2000). An arc or chain may exists between two nodes without the two nodes being dependant.

Minimality (or faithfulness) Condition (Spirtes et al., 2000). Let G be a DAG over V and P be a probability distribution over V. (G, P) satisfies the Minimality condition if and only if for every proper subgraph H of G with vertex set V, (H, P) does not satisfy the Markov condition.

For any Bayesian network (G, P) satisfying the Markov and Minimality conditions, if variables A and B are statistically dependent, then either:

1. There is a directed path in G from A to B; or
2. There is a directed path in G from B to A; or
3. There is a variable C and directed paths in G from C to B and from C to A.

By taking into account assumption A2 and focusing on 2 consecutive levels, only one of the cases 1) and 2) may be considered. In addition, knowing the order of the 2 levels, we are able to say if it is the case 1 or the case 2.

Two nodes belonging to the same level of the cascade arrangement can be statistically dependent in two cases: when the two nodes have a common child by conditioning on this child or when they share a parent. The following theorem allows to deal with these two cases.

Theorem 1. In the cascade arrangement, the nodes belonging to the same level are independent by conditioning on the upper nodes.

Proof. Let us consider two nodes A $\in$ Lx and B $\in$Ly, Lx and Ly being two consecutive levels with Lx >Ly (the upper level is the "parent level" relative to the lower level). If a causal relationships exists between A and B, then A is a parent of B. Let C$\in$ Lx be another parent of B. B is then a collider of the path A – B – C. If Lx is the root level, then A and C are independent. Or else, as C is not a descendant of A, then A and C are independent according to the Markov condition.

Let us consider D$\in$Ly as another child of A. A is a common parent of B and D. Conditioning on A, B and D are independent.

Theorem 2. Let G be a faithful Bayesian Network arranged in cascade and A $\in$ Lx and B $\in$Ly two of its nodes. Lx and Ly being two consecutive levels. If A and B are statistically dependent, then there is an edge between A and B. This node is oriented from the node belonging to upper level towards the node belonging to the lower level.

Proof. If A is a parent of B, as the graph is a causal graph, then A and B are dependent. According to the Markov condition, this direct dependency is represented by an arc in the graph.

The faithfulness condition insures that if there is an arc between A and B, then A and B are dependent. According to the cascade arrangement, and assuming that A belong to the higher level, it is necessary a parent of B.

3.2.2. The causal relationship learning phase

In addition to the two first assumptions, we consider another assumption:

A3: The data set D from which the causal relationship is learned is faithful to the sought network.

Considering these three assumptions and based the two previous theorems, we have developed the causal relationship learning phase of the CBNB algorithm (see Figure 2).

Inputs:

- A Data set D
- The Number of Level L
- The Number of Nodes of each level $\{n_1, n_2, \ldots, n_L\}$
- Threshold α

Output: A Causal Bayesian Network (CBN)

0: Initialize CBN without any arc
1: **for** k =1 to L-1
2: **for** i =1 to n_k
3: **for** j=1 to n_{k+1}
4: Calculate Association $(\boldsymbol{X_i^k};\ \boldsymbol{X_j^{k+1}})$
5: **if** Association $(\boldsymbol{X_i^k};\ \boldsymbol{X_j^{k+1}}) > \alpha$ **then**
6: put an arc from $\boldsymbol{X_i^k}$ to $\boldsymbol{X_j^{k+1}}$
7: **end if**
8: **end for**
9: **end for**
10: **end fo**r
11: return CBN

Figure 2. The CBNB algorithm's causal learning phase

The Association (X, Y) function (line 4). This function calculate the strength of dependency between two variables X and Y. If their dependency is deemed significant (Line 5), an arc is put between them (Line 6). Conditional independence tests such as log-likelihood ratio G^2 (equivalent to Mutual information test) or Pearson's chi-square test (X^2) can be used to measure the strength of dependency. The significance of the dependency is assessed against a threshold for the p-value. The advantage of our algorithm is that it does not measure the dependence of each node with respect to all the remaining variables (as almost all the existing algorithms do). It does so only for a limited number of variables (nodes belonging to the level just below that of the considered node).

Time complexity. The complexity depends on the number of three elementary operations executed by the algorithm: association computation, threshold test and arc adding, if necessary. For two levels k and k +1, $|n_k|*|n_{k+1}|$ associations computations are performed. The same number of threshold tests are made. Finally, at most the same number of arc adding are realized. The association computation operation is as far the most resource-intensive. The complexity evaluation will focus only on this operation. Let us consider $n = \max(n_k, k = 1, \ldots, L-1)$. For each level, the algorithm performs 0 (n^2) association computations. For the whole graph, we will have 0 ((L-1)*n^2) association computations, with L-1 a constant. This corresponds to an order of magnitude of 0 (n^2), where n<<N, with N the total number of variables. By way of comparison, the complexity of the IAMB (Tsamardinos et al., 2003) algorithm is 0 (N^2). As there exists linear algorithm for the strength of dependency calculation, the following theorem can be stated:

Theorem 3. The CBNB algorithm is a polynomial time learning algorithm 0 (n^2), where n << N (N = total number of variables).

4. EXPERIMENTAL STUDIES

The CBNB algorithm was implemented using the bnlearn package for R. All the simulations and the experimental analysis was performed with this package. These studies are divided into two parts. First, we study our algorithm performances according to the threshold ε. Second, we compare our algorithm against some of the best algorithms reported in the literature

As benchmark, we consider three networks (EARTHQUAKE, CANCER and SURVEY) respecting the configuration defined in this work. We start by generating data from conditional probabilities of each network and then we seek to build the network from these data using the tested algorithm. For all experiments, ten sample sizes were considered (100; 200; 300; 500; 800; 1000; 2000; 3000; 5000; 10000). For each sample size, the data generation and experimentation was repeated 5 times and the average of the obtained metrics for the 5 data sets are reported as the result for the considered sample size.

The tests were conducted on PC Intel (R) Core (TM) i5-3340 M CPU @2.70GHz 2.7 GHz, 4Go RAM running Windows 7 Professional.

4.1 Performance metrics

We evaluated performances in terms of the quality of the built network. This evaluation measures the ability of the considered algorithm to provide the real network. Two categories of metric can be distinguished. The first is the posterior probability of the obtained network (BDeu, BIC, etc.) relative to the learning data. The resulting scores measure how the obtained network fit the data. This category is especially useful when the true network is unknown. This is the case for real applications. The second category assesses the equivalency between the obtained network and the true network. As the true network is known in our case (benchmark networks), we used this second category. The performance metrics used are: True Positive (TP) and False Positive (FP). TP is the number of arcs in the obtained (current) network also present in the true (target) network (higher is better). FP is the number of arcs present in the obtained network but not present in the true network (lower is better). For ease of reading, figures have been normalized with respect to the total number of arcs in the true network.

4.2 The CBNB performances assessment

In the implementation of the CBNB algorithm, the association between two nodes is evaluated by performing an independence test by mean of the *ci.test* function from bnlearn. The *mutual information* test statistic for categorical variables was used. We examined the CBNBs' performances (TP, FN) in accordance with the p-value (the target nominal type I error rate) of the test. Large p-value indicates that the dependence relationship is not significant. To decide whether an arc can be put from a node to another, i.e. whether the relationship is significant, a threshold for p-value is fixed. When p-value is

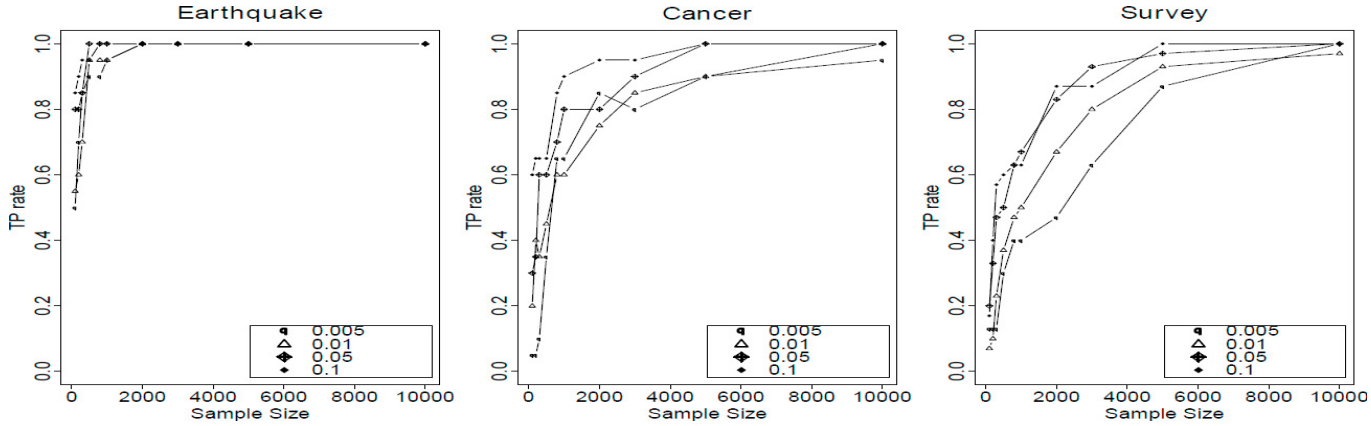

Figure 3. Evolution of TP rate according to the threshold of p-value for the CBNB algorithm

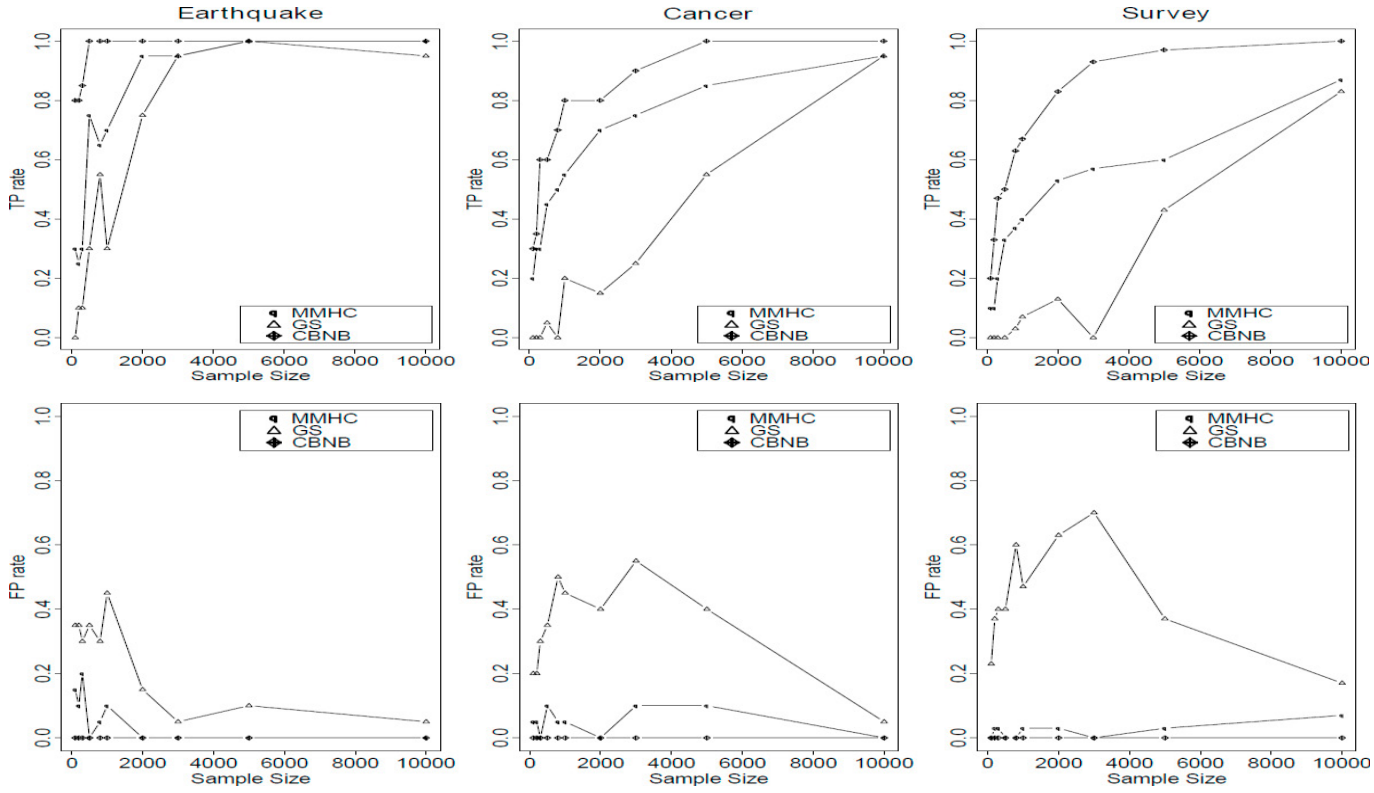

Figure 4. TP and FP rates of GS, MMHC and CBNB algorithms on synthetic data

smaller than the threshold, then the relationship is supposed to be significant. Values α=0.01 or 0.05 are often chosen as significance level for the rejection of the null hypothesis of conditional independence. α= 0.05 is the default threshold in bnlearn. We decide to test different thresholds around this value. For each network, we generate different sample sizes from its true probability distribution. We try to reconstruct the network from the generated data using our algorithm. We compare the obtained network with true ones (known structure). We reported the performances of our algorithm on each sample size and for each value of the threshold. The obtained results are presented in Figure 3.

We observed that for the three networks, by considering each threshold value individually, the TP rate increases as the sample size rise. The difference between the different thresholds lays in the sample size from which the CBNB achieves 100% TP rate. The smaller the threshold, the larger this sample size. But raising the threshold implies an increase of the probability of type I error. This implies an increase of the FP rate. A large p-value threshold, generally beyond 0.1, means no evidence against the null hypothesis (Wasserman, 2004). Therefore, we would suggest to set the threshold to 0.01 if the sample size is large. If the sample size is relatively low, then 0.05 might be set as threshold.

Furthermore, the CBNB produced no FP for the tested threshold values on these three benchmark networks.

4.3 Comparative studies

In this empirical evaluation, we compare the CBNB's performances against those of GS and MMHC in terms of TP and FP. GS (Margaritis, 2003) is an example of constraint-based learning algorithm and MMHC (Tsamardinos et al., 2006) is a hybrid learning algorithm which is able to scale up to thousands of variables.

For this comparative studies, CBNB's threshold for the mutual information p-values was set to 0.05. This is the default value for the target nominal type I error rate in bnlean. For GS and MMHC, their default settings in bnlearn were retained. The results indicate that TP rate obtained by these three algorithms converge to 1. The FP rate is null for CBNB and tends towards zero as the sample size growns for GS and MMHC. As expected, MMHC outperformed GS (see Figure 4). We also observe that the CBNB algorithm outperforms these two state-of-the art algorithms in terms of TP rate and FP rate on these three benchmark networks. CBNB is more sample efficient compared to GS and MMHC as it achieves good results on low sample size. These performances of the CBNB algorithm are due to the incorporation of expert knowledge.

In general, the performances of Bayesian network structural leaning algorithm increase as the sample size grows. As mentioned earlier, in real applications, the number of variables may be very large and of different types. The variables can be machine parameters, material parameters, other process parameters, defects and symptoms. To obtain complete data for the structural learning, each record must have a value for each variable. Furthermore, the data set must contains as least one value of all the levels of all variables. Obtaining such a data set is not easy in practice. Providing a large sample size respecting these constraints is not guarantee. An algorithm allowing to obtain the true network based on a relatively low sample size is therefore very useful for real application.

5. CONCLUSION

Bayesian Networks have interesting characteristics for diagnosis functions development. However, the network structure learning is still a challenging task. Learning BN from data is an NP-hard problem and the performances of existing heuristics depend on the nature and the size of the learning data. The introduction of expert knowledge in the construction process helps to make the structural learning process more effective.

In this paper, we have presented the CBNB algorithm for causal Bayesian network construction. The algorithm is applicable to a particular network with variables arranged in cascade. The algorithm has two phases. The first phase consists of dividing the variables in different levels of causality and defining a complete order among the obtained levels. Then, in the second phase based on unitary traceability data, the existence of dependency between nodes are evaluated. We have performed experimental studies on synthetic data generated from three benchmark networks. The results shown that the CBNB algorithm outperforms the tested two state of the art algorithms (GS and MMHC). It is important to note that CBNB algorithm used strong assumptions namely the possibility to allocate the variables to ordered levels. GS and MMHC do not incorporate this type of knowledge. Results obtained on synthetics data have to be confirmed on real world data provided by our industrial partners of the FUI Traçaverre Project.

Acknowledgments.

This work was supported by Bpifrance (French organization for innovation support and funding, Ministry for Economy, Finance and Industry, and Ministry for higher education and research) through the Traçaverre Project.

REFERENCES

Chen, B., Tavner, P. J., Feng, Y., Song, W. W. & Qiu, Y. N. (2012). Bayesian network for wind turbine fault diagnosis. *EWEA 2012*. Copenhagen, Denmark: European Wind Energy Association.

Cheng, J., Greiner, R., Kelly, J., Bell, D. & Liu, W. (2002). Learning Bayesian networks from data: An information-theory based approach. *Artificial Intelligence,* 137**,** 43-90.

Dey, S. & Stori, J. A. (2005). A Bayesian network approach to root cause diagnosis of process variations. *International Journal of Machine Tools and Manufacture,* 45**,** 75-91.

Diallo, T. M. L., Henry, S. & Ouzrout, Y. (2014). Using Unitary Traceability for an Optimal Product Recall. *In:* Grabot, B., Vallespir, B., Gomes, S., Bouras, A. & Kiritsis, D. (eds.) *Advances in Production Management Systems. Innovative and Knowledge-Based Production Management in a Global-Local World.* Springer Berlin Heidelberg.

Korb, K. B. & Nicholson, A. E. (2003). *Bayesian Artificial Intelligence*, Taylor & Francis.

Margaritis, D. (2003). *Learning Bayesian Network Model Structure from Data.* Doctor of Philosophy, Carnegie Mellon University.

Przytula, K. W. & Thompson, D. (2000). Construction of Bayesian networks for diagnostics. *In:* Aerospace Conference Proceedings, 2000 IEEE,. 193-200 vol.5.

Ramirez, V. J. C. & Piqueras, A. S. (2006). Learning Bayesian Networks for Systems Diagnosis. *In:* Electronics, Robotics and Automotive Mechanics Conference, 2006, Sept.. 125-130.

Riascos, L. A. M., Simoes, M. G. & Miyagi, P. E. (2007). A Bayesian network fault diagnostic system for proton exchange membrane fuel cells. *Journal of Power Sources,* 165**,** 267-278.

Spirtes, P., Glymour, C. N. & Scheines, R. (2000). *Causation, Prediction, and Search*, MIT Press.

Tsamardinos, I., Aliferis, C. & Statnikov, E. (2003). Algorithms for Large Scale Markov Blanket Discovery. *In:* In The 16th International FLAIRS Conference, St. 376-380.

Tsamardinos, I., Brown, L. & Aliferis, C. (2006). The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning,* 65**,** 31-78.

Wasserman, L. (2004). *All of Statistics: A Concise Course in Statistical Inference*, Springer.

Weidl, G., Madsen, A. L. & Israelson, S. (2005). Applications of object-oriented Bayesian networks for condition monitoring, root cause analysis and decision support on operation of complex continuous processes. *Computers & Chemical Engineering,* 29**,** 1996-2009.