

Application possibilities of LBN for civil engineering issues

Agata Siemaszko^{1,*}, and Magdalena Apollo¹

¹Gdańsk University of Technology, Faculty of Civil and Environmental Engineering,
ul. Narutowicza 11/12, 80-233 Gdańsk, Poland

Abstract. Bayesian Networks (BN) are efficient to represent knowledge and for the reasoning in uncertainty. However the classic BN requires manual definition of the network structure by an expert, who also defines the values entered into the conditional probability tables. In practice, it can be time-consuming, hence the article proposes the use of Learning Bayesian Networks (LBN). The aim of the study is not only to present LBN, which can be helpful in civil engineering problems, but also to analyze and evaluate the potential of a selected software. Based on a real example the functional values of the Open Markov, Hugin and AgenaRisk applications were compared.

1 Introduction

A highly competitive and dynamic environment, for the participants of the investment process, forces efficient operation and decision-making [1]. Complex construction projects involving a number of interrelated and interacting variables result in the development of methods and tools supporting the decision-making process. Contemporary methods permanently rooted in solving civil engineering problems include e.g. Artificial Neural Networks (ANNs) [2, 3]. Bayesian networks, allowing not only to capture the issue of data uncertainty, but also to identify interrelations between variables, creating a network of interactions, are still much less common in practice.

Bayesian networks, also known as a causal network or Bayesian Belief Networks (BBN), are a powerful tool for knowledge representation and reasoning under uncertainty, which visually presents the relations between a set of probability variables. In addition, by defining conditional probability tables, the BBN take into account correlation relationships applied in the back propagation mechanism updating *a priori* probability and introducing this way new information in the form of expert knowledge to the network. Updating of the information introduced leads to the construction of the current risk scenario, thus making it possible to bring more reasonable decisions.

Owing the currently used software, the BBN presents probabilistic relationships between a set of variables in a simple and transparent manner [4]. Nevertheless, the classic BBN requires manual definition of the network structure (relations) by an expert, who also

* Corresponding author: agata.siemaszko@pg.edu.pl

defines the values entered into the conditional probability tables [5]. In practice such an action may be quite cumbersome and time-consuming, hence the article proposes the use of Learning Bayesian Networks (LBN), so far rarely used in the field of construction projects. The LBNs employing the available source data are not only able to automatically create a network structure, but also incorporate a formalized mechanism for learning the values of conditional probabilities [5]. The research indicates that this approach not only relieves experts but also eliminates widely commented and criticized subjectivism of Bayesian Networks, which can lead to distortions in the estimation process [6, 7].

The aim of the study is to present Learning Bayesian Networks, which can be helpful in civil engineering problems and also to analyze and evaluate the potential of a selected software. Based on a real example of a problem regarding the operation management of a building facility, the functional values of the Open Markov [8], Hugin [9] and AgenaRisk [10] applications were compared. The authors investigated symptoms related to the technical condition of a building and thus, with the help of LBN, assessed its technical condition.

2 Learning Bayesian Networks (LBN)

According to [11] the BBN consists of two parts:

- qualitative (structural) part: graphical representation of relationships between variables in the form of a graph,
- quantitative (parameter) part: definition of the quantitative dependence between variables in the form of conditional probability tables (calculation of the total probability resulting from relationships between elements).

The BBN is represented by an acyclic graph consisting of a set of vertices and arrows illustrating the nature of the relationship. The vertices represent probabilistic variables while arrows illustrate causal relationships between variables.

The LBNs consist of the same two parts as the traditional BBN: structural and parametric. An important difference, however, is the network building process itself. Eliciting Bayesian networks from experts can be a laborious and difficult procedure especially in the case of large networks. Thus researchers developed methods that could learn the network structure from available data. Furthermore, they formalized methods for learning the conditional probabilities also from input data [5].

The programs the analysis is focused on use three most popular algorithms that allow the automatic construction of the structural part and the parametric LBN. These algorithms include:

- 1) Expectation-Maximisation (EM):
 - algorithm defining the number of iterations, so the number of times the entire data file is used to learn the model parameters,
 - the allowed range of iterations is from 1 to 50, although it is possible that the EM algorithm run less iterations, if the convergence threshold is satisfied earlier (the EM convergence threshold is the difference between the expected log-likelihood of current and previous iterations),
 - the algorithm continues until it either converges or the maximum number of iterations is reached.
- 2) The K2 algorithm belonging to the group of heuristic algorithms, i.e. those that allow to find an approximate solution, and only in specific cases, accurate thanks to the Bayesian evaluation function. This function determines the degree of adjustment of the configuration of a given Xi node to the actual distribution. The K2 algorithm adds

new parents to each node until it improves the value of the matching function or when the maximum number of parents specified by the user is reached.

- 3) The PC algorithm relies on minimizing the amount of necessary d-separation tests, which means quick and effective execution of lower-order tests. In the first phase of the algorithm's operation, the structure of the network is determined, next some of the edges are oriented as far as possible. Thus, a partially oriented structure is generated representing the whole class of potential Bayes networks. Algorithms take two input parameters. First is the confidence level for the independence test. The lower the confidence level, the more attributes are considered independent and a lower number of edges detect the resulting network. The second parameter is the strategy of selecting pairs for the d-separation test. The d-separation test is performed for all attribute pairs in the sample. The complexity of the test for each pair depends on the number of edges eliminated in previous tests.

The algorithms used in the applications proposed in the article are described in more detail in the publications [8, 9, 10]. The subject of this study is simplicity of the problem solution to manage construction work operation.

2.1 Open Markov [8]

OpenMarkov is an open-source software tool for Learning Bayesian Networks (LBN) from data interactively, but also to editing and evaluating several types of several types of PGMs, such as Bayesian networks, influence diagram, as well as to cost-effectiveness analysis

The first method to build a LBN in OpenMarkov is to do it automatically, learning the structure of the network (the directed graph) and its parameters (the conditional probabilities) from a database. The second approach is interactive learning, have an algorithm proposes some modifications of the network, called edits (typically, the addition or the removal of a link), which can be accepted or rejected by the user based on their common sense, their expert knowledge or just their preferences; additionally, the user can modify the network at any moment using the graphical user interface and then resume the learning process with the edits suggested by the learning algorithm. It is also possible to use a model network as the departure point of any learning algorithm, or just to indicate the positions of the nodes in the network learned, or to impose some links.

2.2 Hugin [9]

Hugin software is based on Bayesian networks and influence diagram technology, an advanced artificial intelligence technique widely used for supporting decision-making under uncertainty. The Hugin supports two kinds of LBN: structure learning and parameter learning. Structure learning is the process where the system learns the dependencies between the variables that exist in the data. Parameter learning is the task where you fill in the parameters describing the strength of the dependencies in the learned (or built) structure.

Structure learning in Hugin is supported through the PC-algorithm. Parametric learning is adaptive learning and EM (Estimation-Maximization) learning.

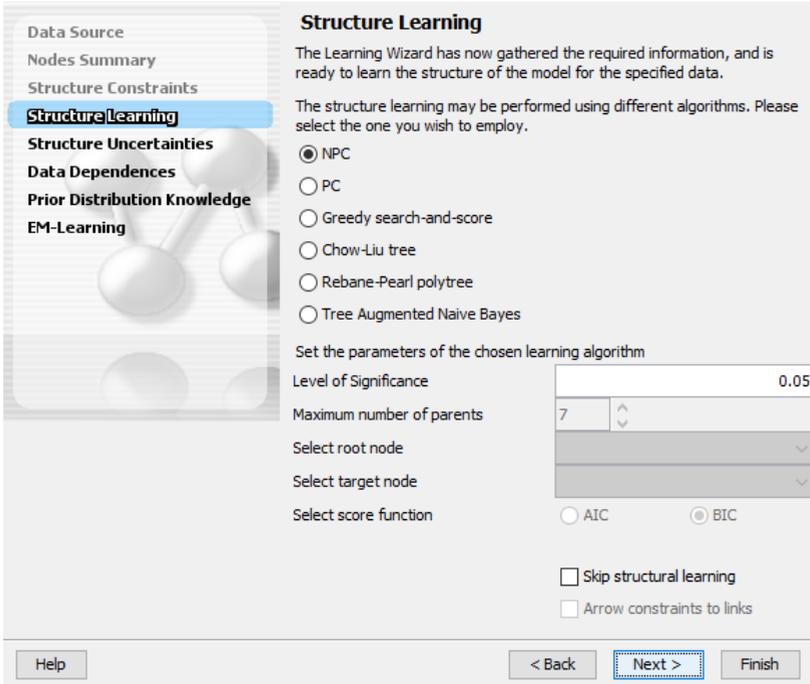


Fig. 1. Structure learning dialog box in Hugin [9].

2.3 AgenaRisk [10]

AgenaRisk enables to automatically learn conditional probability tables using three different ways of table learning:

- learning from data alone,
- learning from data with expert judgement,
- learning from data with expert judgement and custom settings.



Fig. 2. Table learning from data dialog box in AgenaRisk [10].

In all the above mentioned cases learning can be performed even in the case of the so-called “missing data”. Nevertheless, the time necessary to learn a NPT is more a function of the amount of missing data than the amount of data, so more processing time is essential for large data files with lots of missing values. The learning process is performed using an Expectation-Maximisation (EM) algorithm which supports table learning for both Boolean and Labelled types of nodes.

Before running the table learning process it is necessary to prepare and load the data from a file. The dataset should be in “csv” format and if we are not sure what is the format in which the data should be prepared, it is possible to use a button “Generate Example Data File” (Figure 2). This way it is possible to use the generated file as a template.

In the table learning process AgenaRisk also makes it possible to incorporate expert judgement. In this case it is possible to set the ratio of the confidence in knowledge (as represented by the existing node probability tables) compared to the data, from the loaded data file. By default, the application defines a situation where there is no confidence in the knowledge already encoded in the NPTs and a 100% data confidence holds. In this case the existing NPT values will be ignored and the new NPTs will be learnt solely from data.

In the last variant of table learning it is possible to incorporate expert judgement for each node individually.

3 Example problem

The subject of a simplified problem presented on Figure 1 is to determine the current technical condition of a building based on two available observations. The first one (qualitative) concerns the appearance of scratches and cracks on the building's structural walls. There are two states to choose from: YES - scratches or cracks are observed and NO - monitoring jumps into the lack of scratches. The second observation (quantitative) concerns the measurement of heat loss, e.g. by performing the air leak test: BLOWER DOOR TEST. Two states were also taken into account here: YES - the measurement indicates deviations from the assumed norms of air tightness of the facility and NO - no air leaks. Updating the information about the object state, i.e. entering observations into the network, the data is backed up and the probability of occurrence of a specific technical state of the object (GOOD or BAD) is updated.

The main task of the discussed model is to present the possibility of using self-learning Bayesian networks to the problems of managing the operation of buildings by describing the mechanism of learning the structure and parameters of the presented grid in selected programs.

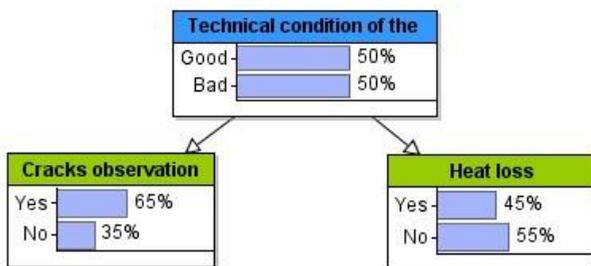


Fig. 3. Problem model: structure and parameters presented in the AgenaRisk program (own study).

3.1 Data for LBN

Bayesian networks can be defined by an expert and /or can learn from the data provided. These data are regarded a set of m vectors ($x_1(i), \dots, x_n(i)$), $i = 1, \dots, m$, generated independently of each other from the probability distribution represented by a "real" Bayesian network (X, S, P) , where S is a structure, and P - parameters. Based on the data loaded into the program, a Bayesian network (X, S', P') is created, which in the assumption should bring as close as possible the output network (X, S, P) . The data used in the network learning process in three alternative programs are presented in Table 1.

Table 1. Data for LBN saved as .csv file (own study).

Technical condition of the building	Cracks observation	Heat loss	Conditional probability
GOOD	YES	YES	$40 \cdot 30 = 1200$
GOOD	YES	NO	$40 \cdot 70 = 2800$
GOOD	NO	YES	$60 \cdot 30 = 1800$
GOOD	NO	NO	$60 \cdot 70 = 4200$
BAD	YES	YES	$90 \cdot 60 = 5400$
BAD	YES	NO	$90 \cdot 40 = 3600$
BAD	NO	YES	$10 \cdot 60 = 600$
BAD	NO	NO	$10 \cdot 40 = 400$

3.2 Problem analysis – Open Markov

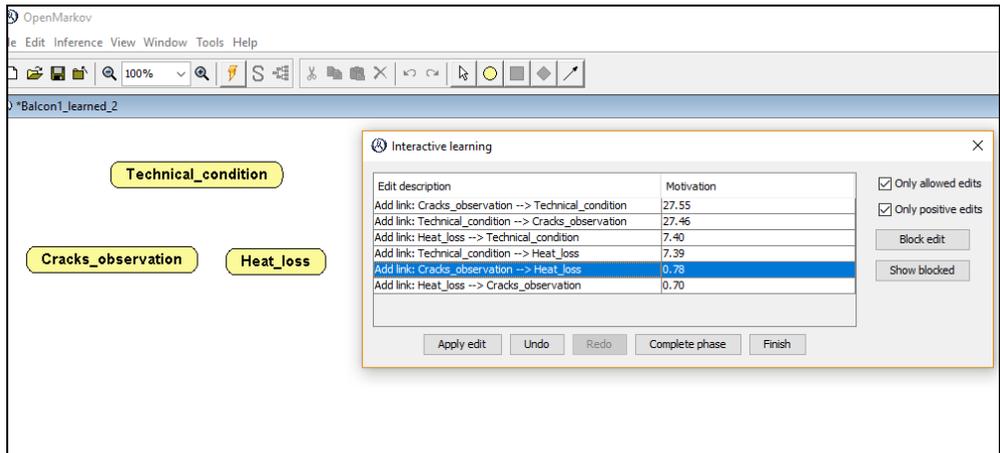


Fig. 4. Problem model in Open Markov: structure and parameters after loading data from .csv file.

The Open Markov program makes it possible to learn the network based on the data contained in the .csv file. Automatic learning does not bring the expected results, because in the constructed network there is a wrong connection of HEAT LOSS and TECHNICAL CONDITION nodes. The solution that gives the best results is interactive network learning. It is based on the fact that the algorithm proposes some network modifications, called changes (adding or deleting a link) that can be accepted or rejected by the user based on common sense, expert knowledge or only preferences. In addition, the user can modify the network at any time using the graphical user interface, and then resume the learning

process in the edition proposed by the learning algorithm. An example of interactive learning is shown in Figure 4.

3.3 Problem analysis – Hugin

The Hugin program turns out to be the easiest to use. During the construction of the network, the user is able to modify all elements starting from the structure of the model, ending with the conditional probabilities tables. The model built from the data contained in the .csv file is identical to the original model and has been shown in the Figure 5. The visual side of the program is exemplary too. The user can easily read all the necessary information from the model.

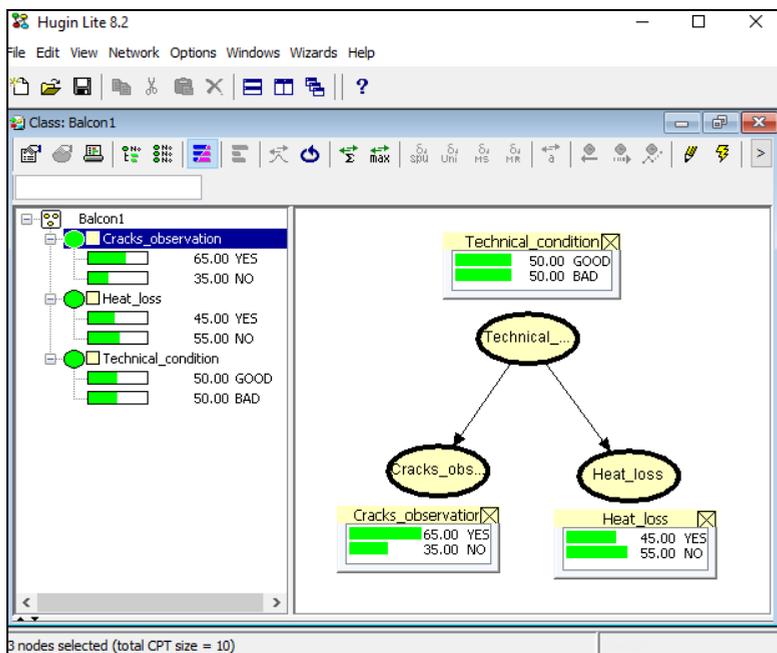


Fig. 5. Problem model in Hugin: structure and parameters after loading data from .csv file.

3.4 Problem analysis – AgenaRisk

According to the information presented in chapter 2.3, the module teaching Bayesian network in AgenaRisk program allows learning the value of conditional probability based on the data provided in the .csv file (the so-called: learning the parameter values from data). A serious limitation of the application possibilities of the program seems to be the inability to automatically build a network structure (the so-called learning the structure from data). What's more, the data exported by AgenaRisk to the .csv file does not include all the necessary combinations of responses, which in other applications are necessary to accurately define the structure. Using the initial model presented in Figure 3, it was possible to load the created .csv file (data presented in table 1). This allowed to verify the value of conditional probabilities, which proved to be in line with the responses of the Open Markov and Hugin programs.

4 Conclusions and recommendations

Based on the conducted analysis, it was shown that only two of the above programs allow automatic network structure creation - Open Markov and Hugin. Consequently, only these applications seem to provide full benefits of the Bayesian self-learning networks. However, in order to prepare the LBN, it is necessary to prepare the data properly. Definition of all available combinations of existing parameters in practice turns out to be very time-consuming, requires a vast theoretical background. Such an operation does not facilitate automatic construction of the network structure. It turns out that insufficient number of rows of data defined in the .csv file, or a possible mistake, may consequently result in improper network construction, and thus erroneous propagation of information. The AgenaRisk is dedicated to learning the values of conditional probabilities only neglecting structure building. In engineering practice, however, automatic parameters learning is much more valuable than the structure itself. The network structure is built intuitively/based on the expert experience. Learning the parameter values, especially in the case of missing data, as well as smooth application functioning are therefore, in the opinion of the authors, the key parameters to be taken into consideration while choosing a program for the LBN. An additional convenience is the visual side and ease of use, and at the same time building the relationship between different variables of the problem. Here, in the authors' opinion, the Hugin program seems to be the best, dialogue operations are very intuitive and all the functions of the program can be easily used without excessive training.

In addition, the article shows that it is possible and desirable to use the LBN to manage the operation of engineering structures. In many publications, it has been proved that Bayesian networks are an effective tool and have been used in various fields so far, including technical diagnostics and medical diagnostics. The success of Bayesian network shows positive results in construction industry, so there is a clear need to apply this methodology to manage the operation of engineering structures.

References

1. B. Grzyl, A. Kristowski, E. Miszewska-Urbańska, *IOP Conf. Ser. Mater. Sci. Eng.* **245**, 8 (2017)
2. S. Arangio, J. L. Beck, *Struct. Control Health Monit.* **19**, 1 (2012)
3. A. Jakubczyk-Gałczyńska, A. Kristowski, R. Jankowski, *Geodetic Congress (BGC Geomatics)* (2017)
4. M. Apollo, *Procedia Eng.* **196** (2017)
5. Neapolitan, R.E., *Learning Bayesian Networks* (Pearson, 2003)
6. S.S. Leu, P.V.H. Son, P.T.H. Nhung, *J. Comput. Civ. Eng.* **29**, 6 (2015)
7. P. C. Liao, H. Shi, Y. Su, X. Luo, *J. Constr. Eng. Manag.* **144**, 3 (2018)
8. OpenMarkov Tutorial - Version 0.2.0 (2017)
9. A. Hanson: Hugin Tutorial (2015)
10. AgenaRisk 7.0 User Manual, Agena Ltd, (2016)
11. M. W. Kembłowski, B. Grzyl, A. Kristowski, A. Siemaszko, *Procedia Eng.* **196** (2017)