

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Continuous Learning of the Structure of Bayesian Networks: A Mapping Study

Luiz Antonio Pereira Silva,
João Batista Nunes Bezerra,
Mirko Barbosa Perkusich, Kyller Costa Gorgônio,
Hyggo Oliveira de Almeida and Angelo Perkusich

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/intechopen.80064>

Abstract

Bayesian networks can be built based on knowledge, data, or both. Independent of the source of information used to build the model, inaccuracies might occur or the application domain might change. Therefore, there is a need to continuously improve the model during its usage. As new data are collected, algorithms to continuously incorporate the updated knowledge can play an essential role in this process. In regard to the continuous learning of the Bayesian network's structure, the current solutions are based on its structural refinement or adaptation. Recent researchers aim to reduce complexity and memory usage, allowing to solve complex and large-scale practical problems. This study aims to identify and evaluate solutions for the continuous learning of the Bayesian network's structures, as well as to outline related future research directions. Our attention remains on the structures because the accurate parameters are completely useless if the structure is not representative.

Keywords: Bayesian network, structure learning, continuous learning, structural adaptation, structural refinement

1. Introduction

Bayesian networks (BNs) are probabilistic graphs used to deal with the uncertainties of a domain [1]. These graphs represent the random variables of this domain and their conditional dependencies. The use of Bayesian networks, also known as Bayesian belief networks, has several points to highlight. Among them, stands out the explicit treatment of

uncertainty, the ease of estimating the state of certain variables given some evidence, as well as having support methods for decision analysis and quick responses by the user [2].

The application domains of Bayesian networks have been extensive [3]. A large number of applications are in the field of medicine [4, 5], being one of the most addressed. There are also applications in the field of forecasting [6], control [7], and modeling for human understanding [8]. In the context of software engineering, fields such as project planning [9], risk management [10], and quality management [11] are addressed. Motivated by the extensive application cited, methods to improve the construction of these graphic models have become a focus of research.

A Bayesian network is defined by a directed acyclic graph (DAG) and a set of parameters for this DAG (NPT). Therefore, in order to build a Bayesian network, the definition of both the graph and the NPT must be considered. Several researches are being carried out with the intention of assisting these definitions [12–14]. However, the solutions proposed are based, for the most part, on the batch process. This process is infeasible in some application domains. Companies, for example, are increasingly storing huge databases with knowledge about their business processes. New knowledge is acquired every time. It is virtually impossible to achieve a highly accurate description of the processes involved without new data being collected or a large amount of data being stored that cannot be analyzed at once. Therefore, the need arose for solutions that continuously incorporate the updated knowledge to prior knowledge.

Ref. [15] investigated the main continuous learning solutions proposed until the development of his study. A comparative analysis between incremental algorithms and an experiment to support this analysis were performed. However, extensions of these, as well as new studies, have since been developed. This chapter aims to describe and analyze existing solutions for continuous learning of Bayesian network structures. A systematic review of the literature is carried out, and the algorithms found are divided into two groups according to their concepts: refinement and structural adaptation. Some guidelines for future research are also described.

2. Learning Bayesian networks

In probability theory, a domain D and its uncertainties can be modeled by a set of random variables $D = \{X_1, \dots, X_n\}$. Each random variable X_i has a set of possible values that combined make up the basis for the modeling of domain D . The occurrence of each possible combination is measured using probabilities that are specified by joint probability distribution, a key concept of probability theory.

In many domains, there is a high number of variables n , requiring the use of probabilistic graphical models for the definition of joint probability distribution. Bayesian networks (BNs) belong to the family of these models that are used to represent a domain and its uncertainties [1]. A BN is a directed acyclic graph (DAG) that encodes a joint probability distribution over a set of random variables D [16]. Formally, a network for D is defined by the pair $B = \{G, \theta\}$.

The first component, G , is a DAG whose vertices correspond to the random variables X_1, \dots, X_n , and the edges represent directed dependencies between variables. The vertices are represented by circles. The edges are represented by arrows indicating the direction of the causal connection between the variables; nevertheless, the information can propagate in any direction in the graph [17].

The chain rule of probability, Eq. (1), can be rewritten as Eq. (2) based on conditional independence rule. Two sets of variables D_x and D_y are independent given D_z if $P(D_x | D_y, D_z) = P(D_x | D_z)$ whenever $P(D_y, D_z) > 0$.

$$P(X_1, \dots, X_n) = \prod_{i=1}^n P(X_i | X_1, \dots, X_{i-1}) \quad (1)$$

$$P(X_1, \dots, X_n) = \prod_{i=1}^n P(X_i | Pa_i) \quad (2)$$

In Eq. (2), Pa_i denotes the set of parents of the variable X_i .

The second component, θ , represents the set of parameters that quantifies the network. This set contains a parameter $\theta_{ijk} = P(X_i = x_i^k | Pa_i = pa_i^j)$ for each possible state x_i^k of X_i , and for each configuration pa_i^j of Pa_i . An example of a Bayesian network is shown in **Figure 1**.

In this case, it is desired to calculate the likelihood of a person having lung cancer given the history of cancer in their family and if this person is a smoker. The node probability tables (NPTs) of the parent nodes represent a prior knowledge of these variables. The NPT of the child node represents the likelihood of a person having cancer given each possible combination of values of the parent nodes.

The goal of the learning process of a Bayesian network is to find a network (or only its structure) that best encodes the joint probability distribution of a domain. Bayesian network learning can be stated as [18]:

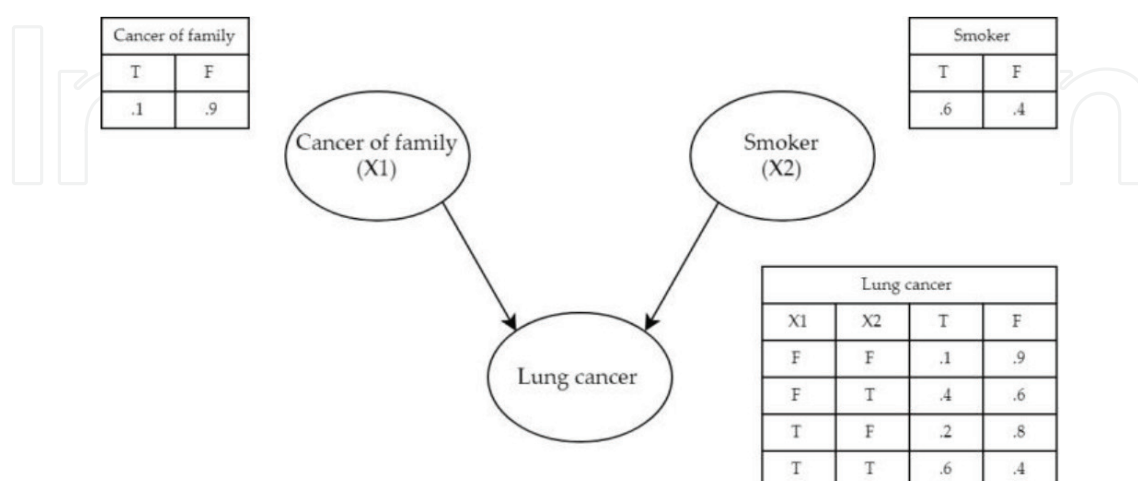


Figure 1. BN example.

Definition 1 (Bayesian network learning): *Given a data set, infer the topology for the belief network that may have generated the data set together with the corresponding uncertainty distribution.*

The learning problem of Bayesian networks can be decomposed into two subproblems: construct the structure, that is, DAG, and define the NPT [19]. Although there are many studies related to the importance of learning NPTs, this study focuses on the first subproblem described. The accurate parameters are completely useless if the structure is not representative. In [20], the importance of the structure of a network in the independence and relevance relationships between the variables concerned is described. Also, in [20], an analysis about the influence of probabilistic networks in the difficulty of representing the uncertainties present in the domain is presented.

The structure can be constructed from data only using machine learning or search techniques, such as those presented in [12, 21–23]. To optimize the definition of structure, the data can be enhanced with expert knowledge. One approach is to consult experts about the posterior probabilities of the structure to reduce the search space, such as presented in [13, 24].

In [25, 26], solutions are presented to complement a Bayesian network with the knowledge of domain experts through the addition of new factors in the model. In this way, it is possible to predict rare events, often not represented in the available databases.

Finally, the structure can be defined only according to the knowledge of specialists, where it is assumed that there are no data available before the structure construction process. In this case, the structure can be defined according to the elicited knowledge of one or multiple experts, as presented in [27, 28].

Most of the solutions to previously reported problems, as well as all of the solutions cited so far, operate as a batch process. The batch process (or batch learning) can be summarized in the delivery of a block of knowledge to an algorithm so that it learns a structure. All the knowledge available to date is used during this process. However, it is inevitable that such information will be inaccurate during the modeling of the domain [29]. For example, if knowledge is acquired from domain experts, the lack of communication between this expert and the expert on graphical models may result in errors in the Bayesian network. Similarly, if the network is being built from a data set, the data set may be inappropriate or inaccurate.

On the other hand, it is neither efficient nor, in some cases, possible to always keep the stored data in search of more representative models using batch learning algorithms [30]. To improve the use of data in the learning problem of Bayesian network structures, it's required solutions that present a continuous process of learning. In the following section, solutions for the continuous learning of the Bayesian network structures are presented.

3. Continuous learning of Bayesian networks' structure

The incentive in the application of processes that realize the learning of Bayesian networks in stages was based, initially, on the observation of the human learning by some researchers. However, the paradigm shift that provided multiple domains generated and stored more and

more data also propelled its development. Continuous (or incremental) learning approaches have some widely accepted definitions found in the literature [31].

In [32], the following precise definition was stated.

Definition 2 (Incremental learner): *A learner L is incremental if L inputs one training experience at a time, does not reprocess any previous experiences, and retains only one knowledge structure in memory.*

In this definition, there are three constraints so that an algorithm can be classified as incremental. In Ref. [32], another definition with a different way of knowledge maintenance was presented.

Definition 3 (Incremental procedure): *A Bayesian network learning procedure is incremental if each iteration l , it receives a new data instance u_l and then produces the next hypothesis s_{l+1} . This estimate is then used by to perform the required task on the next instance u_{l+1} , which in turn is used to update the network and so on. The procedure might generate a new model after some number of k instances are collected.*

This definition relaxes the constraints imposed by Definition 2. For Definition 3, an incremental algorithm can be allowed to process at most k previous instances after encountering a new training instance or to keep k alternative knowledge bases in memory. In [34], another definition is based on Definition 2.

Definition 4 (Incremental algorithm): *An incremental algorithm should meet the following constraints: (i) it must require small constant time per record; (ii) it must be able to build a model using at most one scan of the data; (iii) it must use only a fixed amount of main memory, irrespective of the total number of records it has seen; (iv) it must make a usable model available at any point in time, as opposed to only when it is done with processing the data; and (v) it should produce a model that is equivalent (or nearly identical) to the one that would be obtained by the corresponding batch algorithm.*

Like previous definitions, this definition imposes constraints related to time, memory, and knowledge addressed. The Definition 4 increments the constraint related to the availability of a useful model imposed by Definition 3. Now, a lot due to its application in data streams [34], it is needed to make a usable model available at any point in time, as opposed to only when it is done with processing the data.

Based on the aforementioned definitions of an incremental learning algorithm and in [31], solutions were found that present different learning methodologies. Two groups separate these solutions. The main difference between them is in how they use the acquired knowledge. In one of these groups, denoted by refinement solutions, the data are used according to the knowledge already possessed. This knowledge is maintained in the probabilistic graphic already developed, being only refined with the new data. On the other group, denoted by structural adaptation solutions, the solutions maintain one or more candidate structures and apply to these structures the observations received. This new data set is used to update the sufficient statistics needed to build that candidate structures.

The concepts and information about the type of solutions found in this research are mapped, in an outlined (due to space constraints) and schematic way, in **Figure 2**.

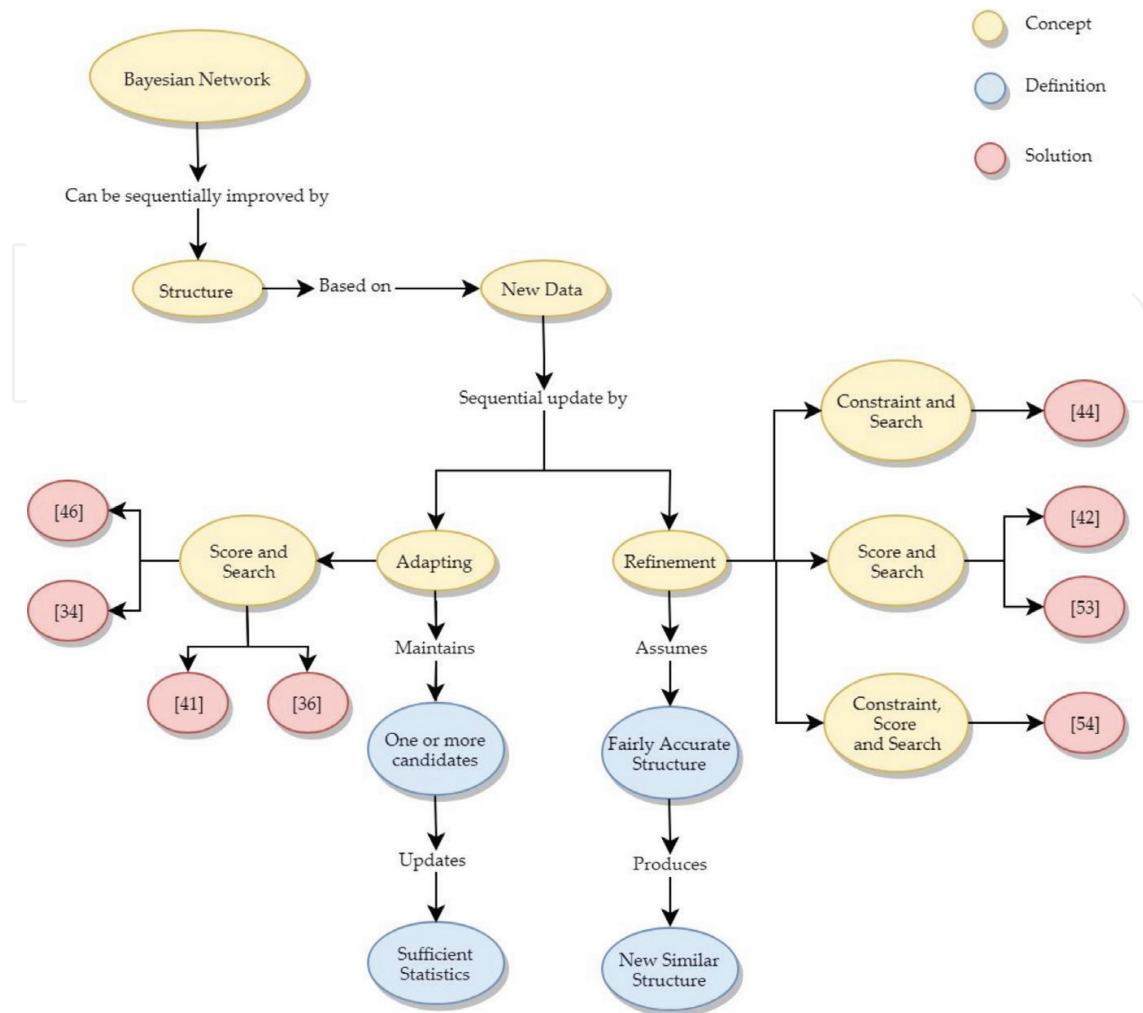


Figure 2. Mind map about solutions.

3.1. Methodology

The continuous learning Bayesian networks structure is kept like an open problem in many application domains. In this study, a systematic literature review is used to identify and evaluate solutions for the continuous learning of the Bayesian networks' structures, as well as to outline related future research directions. A combination of strings was used for title and keyword to identify articles related to continuous learning. Scopus is used as an electronic database.

In the initial search, 4150 items from Scopus were found, but only the first 400 results were checked. This stop was performed because, of these first 400 results, sorted by relevance, a sequence of 150 articles totally unrelated to the search was found. To verify this relationship, three reading steps were performed. Initially, articles were selected considering only the title and abstract. A superficial reading of the remaining articles was then performed. This step consisted of reading and interpreting section titles, figures, graphs, conclusions, and other

elements. In the remaining articles, a critical reading was carried out seeking to interpret and analyze the complete text. The following sections present a description of these efforts.

3.2. Buntine's solution

In [35], a theory refinement-based approach has been proposed. The key task of theory refinement is to update the initial partial theory, usually the expert's prior domain knowledge, as far as new cases produce a posterior knowledge about the space of possible theories, being one of the fundamentals of continuous learning. Given a set of new data and total ordering of the domain variables, the solution updates both the knowledge about the structure and the parameters using different BNs.

For extending and modifying the structure, [35] proposed a batch algorithm that uses the score-and-search-based Bayesian approach. However, using some guidelines presented by the author, it is possible to convert the batch learning into continuous learning process.

The batch algorithm of [35] requires a set of ordered variables $X = \{X_1, \dots, X_n \mid X_i < X_{i+1}, \dots, X_n\}$ according to the prior domain knowledge, where, for the expert, the variables that come first have influence over the others. For each variable X_i , a set of reasonable alternative parent sets $\Pi_i = \{Pa_{i1}, \dots, Pa_{im}\}$ is kept according to some criteria of reasonableness. Each parent set Pa_{ij} is a subset of $\{Y \mid Y < X_i\}$.

A set of alternative parent sets Π_i for the variable X_i is denoted by the parent lattice for X_i . This parent lattice is a lattice structure where subset and superset parent sets are linked together in a web. To access all alternative parent sets $Pa_{ij} \in \Pi_i$ efficiently, only those parent sets with significant posterior probabilities are stored in the parent lattice for X_i . The root node of the parent lattice for X_i is empty set, and the leaves are the sets Pa_{ij} which have no supersets contained in Π_i .

The batch algorithm also requires tree parameters $1 > C > D > E$. These are used to vary the search. The algorithm uses these parameters as base to classify the parent sets as Alive, Asleep, or Dead. The parameter C is used to separate the parent sets that finally take part on the space of alternative networks. The parameter D is used to select the reasonable alternatives to Alive parent sets. The alternatives of Alive parent sets are beams searched by the algorithm. The parameter E is used to select the reasonable alternatives to Dead parent sets. Dead parent sets are alternatives that have been explored and forever determined to be unreasonable alternatives and are not to be further explored. On the other hand, Asleep parent sets are similar but are only considered unreasonable for now and may be made alive later on.

Set the tree parameters C , D , and E to 1 will make the algorithm to be reduce to the K2 algorithm. For this, many researchers cite this algorithm as a generalization of the K2 algorithm [30, 31, 36]. At the end, a structure of alternative networks results from the set of parent sets and the network parameters, denoted by a combined Bayesian network. A pseudo-code for batch algorithm is described in [35].

To convert this batch into continuous learning algorithm, [35] describes two situations that vary according to the time available for the update. In the case where there is a short amount of time for updating the BNs, the algorithm only updates the posterior probabilities of the parent lattices. To this, it is necessary to store posterior probabilities and the counters N_{ijk} for each alternative set of Alive parent sets.

On the other hand, both structure and posterior updates are updated according to new data. For each variable x_i of the combined Bayesian network, it is necessary to: (i) update the posterior probabilities of all alive sets of the lattice, (ii) calculate the new best-posterior, and (iii) expand nodes from the open-list and continue with the search.

The generation of different networks needs to update the posterior probabilities of all alive sets of the lattice. This solution uses sufficient statistics of data that only contains counts of different entries in data instead of data entries, requiring constant time to update sufficient statistics only when new records arrive. Furthermore, [33, 35] performs an additional search over the space of alternative Bayesian networks.

3.3. Friedman and Goldschmidt's solution

Like the previous solution, [34] also addressed the problem of sequential update of the prior domain knowledge. Through the use of sufficient statistics maintained in memory for each network structure at a defined frontier, the knowledge is continuously learned. In this way, this solution provides a method that trades off between accuracy, that is, quality of structure, and storage, that is amount of information about the past observations.

In Ref. [34], three different solutions to sequentially learn BNs have been proposed. Among them, there are two extremes. The naive approach, as it is called, stores all the previously seen data and repeatedly invokes a batch learning procedure after each new observation is recorded. However, despite using as much information as possible, thus increasing the quality of the structure generated, this approach has a high storage cost. In addition, reusing batch learning increases the amount of time and processing spent.

On the other hand, the maximum a posteriori (MAP) probability approach uses a model to store all the information that is considered useful for the next steps in the knowledge update. However, the use of a single model can strongly bias the continuous learning of the model and lose information.

Aware of the disadvantages of previous approaches, [34] presents a new approach, called incremental, which proposes a tradeoff between extremes. The incremental approach does not store all data, unlike the naive approach, and it does not use a single network to represent the prior knowledge, unlike the MAP probability approach. Moreover, it allows flexible choices in the tradeoff between space and quality of the induced networks.

The basic component of this procedure is a module that maintains a set S of sufficient statistics records. The set of sufficient statistics for G , denoted by $Suff(G)$, can be founded by $Suff(G) = \{N_{x_i, Pa_i} : 1 \leq i \leq n\}$. Similarly, given a set S of sufficient statistics records, the set of network structures, denoted by $Nets(S)$, can be evaluated using the records in S by $Nets(S) = \{G : Suff(G) \subseteq S\}$.

Two structures can easily keep track by maintaining a slightly larger set of statistics, for example, suppose on the deliberating choice between two structures G and G' . To evaluate G , in order to use a scoring function, it is needed to maintain the set $Suff(G)$. On the other hand, to evaluate G' , it is needed to maintain the set $Suff(G')$. Now, supposing that G and G' differ only by one arc from X_i to X_j , note a large overlap between $Suff(G)$ and $Suff(G')$. Namely, $(G) \cup Suff(G') = Suff(G) \cup \{N_{X_j, Pa_j}\}$, where Pa_j is the parent set of Y in G' . That argument can be useful when one considers the use of the greedy hill climbing search procedure, for example. Note that it is possible to evaluate the set of neighbors of s by maintaining a bounded set of sufficient statistics.

Generalizing this discussion, the incremental approach can be applied to any search procedure that can define a search frontier. This frontier, denoted by F , consists of all the networks it compares in the next iteration. The choice of F determines which sufficient statistics are maintained in memory. After a new instance is received (or, in general, after some number of new instances are received), the procedure uses the sufficient statistics in s to evaluate and select the best scoring network in the frontier F or in $Nets(s)$. A pseudo-code for incremental approach is described in [34].

When this approach is instantiated with the greedy hill climbing procedure, the frontier F consists of all the neighbors of B_n . With beam search, on the other hand, the frontier F consists of all j candidates.

Many scoring functions can be used to evaluate the “fitness” of networks with respect to the training data and then to search for the best network. However, the incremental approach collects different sufficient statistics in different moments of the learning process. Thus, they need to compare Bayesian networks with respect to different data sets. This problem happens because, unlike [35, 34] may consider those structures that, previously, were considered as non-promising (the ones that were out of the frontier).

The two main scoring functions commonly used to learn Bayesian networks, Bayesian scores [37] and minimal description length (MDL) [38], are inappropriate for this problem. In order to overcome this problem, [34] proposed an averaged MDL measure $S'_{MDL}(G|D) = S_{MDL}(G|D)/N$, where N is the number of instances of the data set. This score measures the average encoding length per instance.

Analyzing both [35, 34] like hill-climbing searchers, they perform, for each node, operations to increase the score of the resulting structure, without introducing a cycle into the network and based on the assumption that they start from an arc-less network, as can be observed in their pseudo-code. Both stop when performing a single operation cannot increase the network's score. The difference between [35, 34] is the neighborhood composition. While [35] uses only the addition operator to construct neighbors, [34] uses the addition, reversion, and deletion of an arc.

3.4. Roure's solution

The conversions of the batch learning approaches present in [35, 34] in continuous learning approaches paved the way for popular batch algorithms like the B, K2 [37] and HCMC [39] algorithms to be turned into incremental ones [31].

In Ref. [40], two heuristics to change a batch hill-climbing search (HCS) into an incremental hill-climbing search algorithm based on combining of sufficient statistics with reduced search space have been proposed. In the batch version of HCS algorithm, a search on the space called neighborhood is performed to examine all possible local changes that can be made in order to maximize the scoring function.

Similar to the frontier presented in [34], a neighborhood of a model B consists of all models that can be build using one or more operators of a set of operators $OP = \{ \text{"Add Edge"}, \text{"Delete Edge"}, \text{"Reverse Edge"} \}$ and argument pairs A . Taking that into account, the sequence of operators and argument pairs added to obtain the final model B_f can be denoted by search path. Let B_0 be an initial model, a final model obtained by a hill-climbing search algorithm can be described by $B_f = op_n(\dots(op_1, A_1), \dots), A_n)$, where the search path $O_{op} = \{(op_1, A_1), \dots, (op_n, A_n)\}$ was used to build B_f .

The heuristics presented in [40] are based on two main problems: when and which part to update, and how to calculate and store sufficient statistics. The first heuristic is called by traversal operators in correct order (TOCO). TOCO verifies the already learned model and its search path for new data. If the new data alter the search path, then it is worth to update an already learned model. The second heuristic is called by reduced search space (RSS). RSS identified when the current structure needs to be revised. At each step of the search path, it stores top k models in a set B having the score close to the best one. The set B reduces the search space by avoiding to explore those parts of the space where low-quality models were found during former search steps. A pseudo-code for incremental hill-climbing search algorithm is described in [40].

3.5. Lam and Bacchus's solution

In [41], another continuous learning solution based on an extension of batch solution is presented. The batch solution used as base is presented in [42]; however, it will not be presented here because the new solution is not coupled to their batch algorithm. The proposed extension aims to perform a review of the BN structure incrementally as new data about a subset of variables were available.

This revision is done using the structure of the BN as prior probability under the implicit assumption that the existent network is already a fairly accurate model of the database. This assumption is a way to incorporate domain knowledge into the problem; however, the new refined network structure should be similar to the existing one, skewing the process.

The solution of [41] also proved, like [35] and based on the MDL measure, that if partial network structure of the whole structure gets, by changing its topology, better score to scoring function, then the whole network structure be improved if no cycles are introduced. Based on this, [41] developed an algorithm to update the BN by improving parts of it. This algorithm produces a new partial network structure based on new data set and the existing network using an extension of MDL. It then locally modifies the old structure comparing and changing correspond part according to new partial network.

The source data to algorithm consists of two components: the new data and the existent network structure. Considering the MDL principle states, finding a partial network G_p is a must that minimizes the sum of the of the length of the encoding of: (i) the partial network G_p , (ii) the

new data given the network G_p , and (iii) the existent network given the network G_p . To calculate the encoding length of the first two items, there are guidelines in [41]. To calculate the length of the encoding of the third item, it is needed to compute the description of the complete existent network G given the network G_p , that is, to describe the differences between G and G_p . These differences are described by: (i) a listing of reversed arcs, (ii) the additional arcs of G , and (iii) the missing arcs of G .

A simple way to encode an arc is to describe the source node and the destination node. $2 \log n$ bits are required to describe an arc, since is required $\log n$ to identify one, provided that exists n nodes. Let r , a , and m be, respectively, the number of reversed, additional, and missing arcs in G with respect G_p , the description length G given the network G_p is $(r + a + m) 2 \log n$.

In order to learn the local structure, the batch algorithm proposed by [42] or other algorithm using the scoring function for each node of the partial structure is presented by Eq. (3).

$$DL_i = |Pa_i| + \log n + \sum_{X_j \in Pa_i} I(X_i; X_j) + (r_i + a_i + m_i) 2 \log n \quad (3)$$

With the third term of the equation, [41] avoided using the sufficient statistics of the old data.

After the new partial structure is learned, the review process continues with the attempt to obtain a refined structure of lower total description length with the aid of the existent structure G and the partial structure G_p . The review problem now is reduced to choosing appropriate subgraphs, denoted by the marked subgraph [41], for which we should perform parent substitution in order to achieve a refined structure of lowest total description length.

In an attempt to avoid creating cycles during each subgraph substitution, [41] uses best-first search to find the set of subgraph units that yields the best reduction in description length without generating any cycles. In addition, a list $s = \{s_1, \dots, s_n\}$ containing a ranking of all subgraphs in ascending order of the benefit gained.

3.6. Shi and Tan's solution

In [43], an efficient hybrid incremental learning algorithm is proposed. All solutions presented so far are score-and-search-based solutions. This solution consists of a polynomial-time constraint-based technique and a hill-climbing search procedure. In this way, this solution provides a hybrid algorithm that offers considerable computational complexity savings and slightly better model accuracy.

The first fragment that composes the solution is based on a constraint-based technique. The purpose of this technique is to select candidate parent set for each variable on data. For each variable X_j , a candidate parents set S_{X_j} is set up containing all the other variables at first. If variable X_j was independent from X_i conditioned on some variables set C in previous learning procedures, the algorithm reperforms the conditional independent test and remove X_j from S_{X_j} if the independence still holds.

After this, a heuristic procedure called HeuristicIND is proposed to reduce S_{X_j} further. This procedure tries to find out a variable set to separate X_i and X_j conditionally. Using the current

network structure, a tree-shaped undirected skeleton is then built up using [44]. The pseudo-codes for this procedure and for the polynomial-time constraint-based technique and a hill-climbing search procedure are described in [43].

3.7. Others continuous learning solutions

In [45], an improvement in the refinement process of [34] is presented. In this study, an incremental method for learning Bayesian networks based on evolutionary computing, denoted by IEMA, is developed.

The solutions presented so far assume that a stationary stochastic process produces all knowledge, that is, the ordering of the database is inconsequential. However, in many application domains of BNs, such as financial problems [46], the processes vary according to the time, the data are non-stationary or piecewise stationary distributed, which would reduce the adequacy of the solutions already mentioned. In [47], the assumption on stationary data is relaxed and an incremental learning Bayesian network based on non-stationary data domains is developed.

In [48], the streaming data prediction is addressed. A parallel and incremental solution for learning of BNs from massive, distributed, and dynamically changing data by extending the classical scoring and search algorithm and using MapReduce is presented.

Ref. [37] also presents an algorithm for stream and online data, more precisely, data that are privately and horizontally shared among two or more parties. This algorithm is based on an efficient version of sufficient statistics to learning privacy-preserving Bayesian networks.

In [49], an active and dynamic method of diagnosis of crop diseases has been proposed based on Bayesian networks and incremental learning. To incremental learning, a new algorithm for dynamically updating the Bayesian network-based diagnosis model over time also is proposed.

Ref. [50] transformed the local structure identification part of Max-min Hill-climbing (MMHC) algorithm into an incremental fashion by using heuristics and applied incremental hill-climbing to learn a set of candidates-parent-children for a target variable.

In [51], an incremental algorithm for BN structure learning that can deal with high dimensional domains has been proposed.

In [52], the concept of influence degree is used to describe the influence of new data on the existing BN. A scoring-based algorithm for revising a BN iteratively by hill-climbing search for reversing, adding, or deleting edges has been proposed.

In [53], an approach to incremental structure optimize is presented. Based on a specific method, this approach decomposes the initial network into several subnets created from a junction tree developed using information about the joint probability of the network. With some adaptations, it can be used as a continuous learning algorithm.

It is also important to point out that studies are found that denominate their algorithm as incremental, but these algorithms result in incremental learning of variables or small parts of the network as they are available and do not necessarily generate a new model, as it is constrained in the definition.

3.8. Comments and future research

Some comments comparing some solutions have already been made during your descriptions. However, to facilitate the understanding of the methodologies used, some techniques and characteristics were compared below. A summary table among some solutions is presented in **Table 1**.

Some features of the solutions were discussed in **Table 1**. These features are important for the differentiation of the proposal of each algorithm. It is noteworthy that, among the outstanding solutions, none proposes to use the domain specialist as a source of posterior knowledge, only as a source of prior knowledge. This knowledge can be increased over time and can be used to improve the built network. In addition, changes that are not able to be identified only with the use of data, such as adding factors and incomprehensibility of the model, can be identified.

When checking the stochastic process, it is noted that only [47] adopts a non-stationary domain, even considering the increasing diversity of domain. The other quoted solutions kept their focus on stationary domains, more common at the time of their developments. Still considering the domain, the effectiveness of the algorithms is rarely validated in a real domain, even if it is done with experiments that use data coming from real domains.

The local search is a standard present among the search procedures used. Despite their high computational complexity, methods were developed so that this was not a constraint and

Solution	Domain expert	Type of structure changes	Scoring function	Search procedure	CI	Stochastic process
[36]	Prior knowledge	Adding edges	Any	Any local search	—	Stationary
[34]	Prior knowledge	Adding, deleting, or reversing edges	Averaged MDL	Any local search	—	Stationary
[40]	Prior knowledge	Adding, deleting, or reversing edges	Any	Hill-climbing search	—	Stationary
[41]	Prior knowledge	Adding, deleting, or reversing edges	Extension of MDL	Best-first search	—	Stationary
[43]	Prior knowledge	Adding, deleting, or reversing edges	—	Hill-climbing search	InfoChi	Stationary
[47]	Prior knowledge	Adding, deleting, or reversing edges	—	Any local search	—	Non-stationary

Table 1. Summary table among some solutions.

they then continued to be used. Only one solution made use of conditional independence (CI) tests. Ref. [43] developed a new technique that is used as the basis for CI tests. Considering scoring functions, some solutions leave open the use of any function. However, an adaptation to their applications is necessary in sufficient statistics used in some, for example, in [34]. On the other hand, some solutions use the MDL measure, but already adapted, either to reduce computational complexity or to achieve better results in different data set data. Other features can be addressed in future reviews, such as computational complexity, procedure focus, and the type of application domain, among others.

4. Conclusions

This chapter presents a survey based on a systematic literature review of continuous learning solutions of Bayesian network structures. It searches articles with an algorithm or approach considered as incremental according to the definitions presented in the text.

The solutions found can be classified as structural adaptation or refinement. The first group, in short, uses the new data set to maintain sufficient statistics that are used to store the existing knowledge about the domain. The second group uses this new set of data to perform a refinement in the network, based on previous knowledge coupled in the old structure.

Finally, the presence of the posterior knowledge of the domain specialist during the incremental learning process and experiments in real domains for the validation of some solutions findings are in the future works.

Acknowledgements

The authors would like to thank Federal University of Campina Grande in Brazil for supporting this study.

Conflict of interest

The author declares that there is no conflict of interest regarding the publication of this chapter.

Author details

Luiz Antonio Pereira Silva, João Batista Nunes Bezerra, Mirko Barbosa Perkusich*, Kyller Costa Gorgônio, Hyggo Oliveira de Almeida and Angelo Perkusich

*Address all correspondence to: mperkusich@gmail.com

Federal University of Campina Grande, Paraíba, Brazil

References

- [1] Ben-Gal I. Bayesian Networks. Encyclopedia of Statistics in Quality and Reliability. John Wiley and Sons; 2008
- [2] Lee E, Park Y, Shin JG. Large engineering project risk management using a Bayesian belief network. Expert Systems with Applications. 2009;**36**(3):5880-5887
- [3] Daly R, Shen Q, Aitken S. Learning Bayesian networks: Approaches and issues. The Knowledge Engineering Review. 2011;**26**(2):99-157
- [4] Lucas PJF, Van der Gaag LC, Abu-Hanna A. Bayesian networks in biomedicine and health-care. Artificial Intelligence in Medicine. 2004;**30**(3):201-214
- [5] Shwe M, Cooper G. An empirical analysis of likelihood-weighting simulation on a large, multiply connected medical belief network. Computers and Biomedical Research. 1991;**24**(5):453-475
- [6] Abramson B et al. Hailfinder: A Bayesian system for forecasting severe weather. International Journal of Forecasting. 1996;**12**(1):57-71
- [7] Forbes J et al. The batmobile: Towards a Bayesian automated taxi. IJCAI. 1995;**95**:1878-1885
- [8] Husmeier D. Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks. Bioinformatics. 2003;**19**(17):2271-2282
- [9] Pendharkar PC, Subramanian GH, Rodger JA. A probabilistic model for predicting software development effort. IEEE Transactions on Software Engineering. 2005;**7**:615-624
- [10] Fan CF, Yu YC. BBN-based software project risk management. Journal of Systems and Software. 2004;**73**(2):193-203
- [11] Jeet K, Bhatia N, Minhas RS. A Bayesian network based approach for software defects prediction. ACM SIGSOFT Software Engineering Notes. 2011;**36**(4):1-5
- [12] Neapolitan RE. Learning Bayesian networks. Vol. 38. Upper Saddle River, NJ: Pearson Prentice Hall; 2004
- [13] Heckerman D. A tutorial on learning with Bayesian networks. In: Learning in Graphical Models. Dordrecht: Springer; 1998. pp. 301-354
- [14] O'Hagan A et al. Uncertain Judgements: Eliciting Experts' Probabilities. Chichester: John Wiley & Sons; 2006
- [15] Huang H et al. A comparatively research in incremental learning of Bayesian networks. Intelligent Control and Automation. 2004. WCICA 2004. Fifth World Congress on. Vol. 5. IEEE, 2004
- [16] Friedman N, Geiger D, Goldszmidt M. Bayesian network classifiers. Machine Learning. 1997;**29**(2-3):131-163

- [17] Amari S. The Handbook of Brain Theory and Neural Networks. London, England: MIT Press; 2003
- [18] Sangüesa R, Cortés U. Learning causal networks from data: A survey and a new algorithm for recovering possibilistic causal networks. *AI Communications*. 1997;**10**(1):31-61
- [19] Neil M, Fenton N, Nielson L. Building large-scale Bayesian networks. *The Knowledge Engineering Review*. 2000;**15**(3):257-284
- [20] Druzdel MJ, Van Der Gaag LC. Building probabilistic networks: Where do the numbers come from? *IEEE Transactions on Knowledge and Data Engineering*. 2000;**12**(4):481-486
- [21] van Dijk S, Van Der Gaag LC, Thierens D. A skeleton-based approach to learning Bayesian networks from data. *European Conference on Principles of Data Mining and Knowledge Discovery*; Berlin, Heidelberg: Springer; 2003
- [22] Tsamardinos I, Brown LE, Aliferis CF. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*. 2006;**65**(1):31-78
- [23] Zhang Y, Zhang W, Xie Y. Improved heuristic equivalent search algorithm based on maximal information coefficient for Bayesian network structure learning. *Neurocomputing*. 2013;**117**:186-195
- [24] Heckerman D, Geiger D, Chickering DM. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*. 1995;**20**(3):197-243
- [25] Constantinou AC, Fenton N, Neil M. Integrating expert knowledge with data in Bayesian networks: Preserving data-driven expectations when the expert variables remain unobserved. *Expert Systems with Applications*. 2016;**56**:197-208
- [26] Constantinou A, Fenton N. Towards smart-data: Improving predictive accuracy in long-term football team performance. *Knowledge-Based Systems*. 2017;**124**:93-104
- [27] Hu X-X, Wang H, Shuo W. Using expert's knowledge to build Bayesian networks. *Computational Intelligence and Security Workshops, 2007. CISW 2007. International Conference on*. IEEE; 2007
- [28] Richardson M, Domingos P. Learning with knowledge from multiple experts. In: *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*; 2003
- [29] Lam W. Bayesian network refinement via machine learning approach. *IEEE Transactions on Pattern Analysis & Machine Intelligence*. 1998;**3**:240-251
- [30] Zeng Y, Xiang Y, Pacekajus S. Refinement of Bayesian network structures upon new data. *International Journal of Granular Computing, Rough Sets and Intelligent Systems*. 2009;**1**(2):203-220
- [31] Alcube JR. Incremental methods for Bayesian network structure learning. *AI Communications*. 2005;**18**(1):61-62

- [32] Langley P. Order effects in incremental learning. In: *Learning in Humans and Machines: Towards an Interdisciplinary Learning Science*. Vol. 136. Pergamon; 1995. p. 137
- [33] Friedman N, Goldszmidt M. Sequential update of Bayesian network structure. In: *Proceedings of the Thirteenth conference on Uncertainty in artificial intelligence*; Morgan Kaufmann Publishers Inc.; 1997
- [34] Domingos PM, Hulten G. Catching up with the data: Research issues in mining data streams. *DMKD*. 2001
- [35] Buntine W. Theory refinement on Bayesian networks. In: *Proceedings of the Seventh conference on Uncertainty in Artificial Intelligence*; Morgan Kaufmann Publishers Inc.; 1991
- [36] Samet S, Miri A, Granger E. Incremental learning of privacy-preserving Bayesian networks. *Applied Soft Computing*. 2013;**13**(8):3657-3667
- [37] Cooper GF, Herskovits E. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*. 1992;**9**(4):309-347
- [38] Lam W, Bacchus F. Learning Bayesian belief networks: An approach based on the MDL principle. *Computational Intelligence*. 1994;**10**(3):269-293
- [39] Kočka T, Castelo R. Improved learning of Bayesian networks. In: *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*; Morgan Kaufmann Publishers Inc.; 2001
- [40] Alcobé JR. Incremental hill-climbing search applied to Bayesian network structure learning. In: *Proceedings of the 15th European Conference on Machine Learning*; Pisa, Italy; 2004
- [41] Lam W, Bacchus F. Using new data to refine a Bayesian network. *Uncertainty Proceedings*. 1994;**1994**:383-390
- [42] Castelo R, Kocka T. On inclusion-driven learning of Bayesian networks. *Journal of Machine Learning Research*. 2003;**4**:527-574
- [43] Shi D, Tan S. Incremental learning Bayesian network structures efficiently. In: *Control Automation Robotics & Vision (ICARCV), 2010 11th International Conference on*. IEEE; 2010
- [44] Chow CK, Liu CN. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*. 1968;**14**:462-467
- [45] Tian F et al. Incremental learning of Bayesian networks with hidden variables. In: *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*. IEEE; 2001
- [46] Shi D, Tan S. Incremental learning bayesian networks for financial data modeling. In: *Intelligent Control, 2007. ISIC 2007. IEEE 22nd International Symposium on*. IEEE; 2007

- [47] Nielsen SH, Nielsen TD. Adapting Bayes network structures to non-stationary domains. *International Journal of Approximate Reasoning*. 2008;**49**(2):379-397
- [48] Yue K et al. A parallel and incremental approach for data-intensive learning of Bayesian networks. *IEEE Transactions on Cybernetics*. 2015;**45**(12):2890-2904
- [49] Zhu Y et al. Mathematical modelling for active and dynamic diagnosis of crop diseases based on Bayesian networks and incremental learning. *Mathematical and Computer Modelling*. 2013;**58**(3-4):514-523
- [50] Yasin A, Leray P. iMMP: a local search approach for incremental Bayesian network structure learning. *International Symposium on Intelligent Data Analysis*; Berlin, Heidelberg: Springer; 2011
- [51] Yasin A, Leray P. Incremental Bayesian network structure learning in high dimensional domains. In: *Modeling, Simulation and Applied Optimization (ICMSAO)*, 2013 5th International Conference on. IEEE; 2013
- [52] Liu W et al. A Bayesian network-based approach for incremental learning of uncertain knowledge. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*. 2018;**26**:87-108
- [53] Chunsheng G, Qiquan S. Incremental structure optimize of Bayesian network based on the lossless decomposition. In: *Artificial Intelligence and Computational Intelligence (AICI)*, 2010 International Conference on. Vol. 2. IEEE; 2010