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

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX
INDEXED

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

Building a Bayesian Network Model Based on the Combination of Structure Learning Algorithms and Weighting Expert Opinions Scheme

Wichian Premchaiswadi and Nipat Jongsawat
Graduate School of Information Technology in Business, Siam University
Thailand

1. Introduction

Bayesian networks (BNs) is probabilistic graphical models that are widely used for building expert systems in several application domains. In the context of expert systems, either probabilistic or heuristic, the development of explanation facilities is important for three main reasons. First, the construction of those systems with the help of human experts is a difficult and time consuming task, and prone to errors and omissions. A Bayesian network tool can help the knowledge engineers and experts who are taking part in the project to debug the system when it does not yield the expected results and even before a malfunction occurs. Second, human beings are reluctant to accept the advice that is offered by a machine if they are not able to understand how the system arrived at those recommendations. Third, an expert system that is used as an intelligent tutor must be able to communicate to the apprentice the knowledge it contains, the way in which the knowledge has been applied to arrive at a conclusion, and what would have happened if the user had introduced different pieces of evidence (what-if reasoning). One of the most difficult obstacles in the practical application of probabilistic methods is the effort that is required for model building and, in particular, for quantifying graphical models with numerical probabilities. The construction of Bayesian Networks (BNs) with the help of human experts is a difficult and time consuming task, which is prone to errors and omissions especially when the problems are very complicated or there are numerous variables involved. Learning the structure of a BN model and causal relations from a dataset or database is important for extensive BNs analysis. In general, the causal structure and the numerical parameters of a BN can be obtained using two distinct approaches. First, they can be obtained from an expert. Second, they can also be learned from a data set. The main drawback of the first approach is that sometimes there is not enough causal knowledge to establish the structure of the network model with certainty and estimation of probabilities required for a typical application is a time-consuming task because of the number of parameters required (typically hundreds or even thousands of values). Thus, the second approach can initially help human experts or a group of experts build a BN model and they can make it applicable at a later time. In practice, some combination of these two approaches is typically used.

This article presents a SMILEBN web application for building a Bayesian network model. The SMILEBN can build a BN model based on using two techniques: 1) to build a BN model by applying the structure learning algorithms to a dataset, and 2) to use group decision making technique for weighting the degree of an expert's opinion in identifying influential effects from parent variables to child variables in the model. Finally, the BN model which all the experts agree to use is obtained. In case that the BN model which is built from a data set is complex, the SMILEBN users can set a threshold value for the model in order to minimize the number of relationships among the nodes in the BN model. When the number of relationships among the nodes decreases, the complexity of the conditional probability table on each child node also decreases.

This article is organized as follows: Section 2 addresses related work. Section 3 presents the tools that are used to build a BN causal structure from a dataset. Section 4 presents the method to use group decision making technique for weighting the degree of an expert's opinion in identifying influential effects from parent variables to child variables. Section 5 presents a SMILEBN web application. Section 6 presents a conclusion and discusses some perspectives and ideas for future work.

2. Related work

There are various kinds of software applications that can be used to create decision theoretic models, learn the causal structure, and perform diagnosis based on BNs. There are both commercial and non-commercial software applications available. The commercial software applications are widely used in a business environment. Many of them are integrated into business analysis software and used particularly for solving difficult business problems. The non-commercial software applications are extensively used for the educational purposes. This article reviews only the most relevant subset of non-commercial software applications based on BNs.

B-Course is an analysis tool that was developed in the fields of Bayesian and causal modelling (Mylltmaki et al., 2002). It is a free web-based online data analysis tool, which allows users to analyze data for multivariate probabilistic dependencies. It also offers facilities for inferring certain type of causal dependencies from the data. B-Course is used via a web-browser, and requires the user's data to be a text file with data presented in a tabular format typical for any statistical package (e.g., SPSS, Excel text format). It offers a simple three step procedure (data upload, model search, and analysis of the model) for building a BN dependency model. After searching the model, B-Course provides the best model to the user via a report. Users can continue to search for the next best model but they must make the decision for selecting the best model that fits their needs. Selecting the best model is sometimes very difficult for inexperienced users. In B-Course, there are no structural learning algorithms provided for the user to aid in selection. The analysis method, modelling assumptions, restrictions, model search algorithms, and parameter settings are totally transparent to the user.

Elvira is a tool for building and evaluating graphical probabilistic models (Lacave et al., 2007). It is a non web-based application. It is implemented in Java, so that it can run on different platforms. It contains a graphical interface for editing networks, with specific options for canonical models (e.g., OR, AND, MAX, etc.), exact and approximate algorithms

for discrete and continuous variables, explanation facilities, learning methods for building networks from databases, algorithms for fusing networks, etc. Elvira is structured as four main modules: (1) data representation- containing the definition of the data structures that are needed for managing BNs and IDs in Java, (2) data acquisition- including the classes that are necessary for saving and loading a network from either a file or a database, (3) processing - implementing the algorithms for processing and evaluating models, and (4) visualization - defining the Elvira graphical user interface (GUI) which obviously makes use of the classes that are included in the previous modules.

GeNIe (Graphical Network Interface) is a versatile and user friendly development environment for building graphical decision models (Druzdzel, 1999). The original interface was designed for a Structural Modeling, Inference, and Learning Engine (SMILE). GeNIe may be seen as an outer shell to SMILE. GeNIe is implemented in Visual C++ and draws heavily on the Microsoft foundation classes. GeNIe provides numerous tools for users such as an interface to build Bayesian network models or influence diagrams, to learn the causal relationships of a model using various algorithms, and to perform model diagnosis. In order to use GeNIe efficiently, the GeNIe software must be installed and the user should have some background knowledge about probabilistic graphical models and become familiar with the tools provided in GeNIe.

Poompuang, et al presents a development environment for building graphical decision-theoretic models based on BNs and influence diagrams working on the website by utilizing an original engine called "SMILE" (Poompuang, et al., 2007). They propose the idea of building and developing graphical decision-theoretic models on a web page in order to overcome such the limitation of Bayesian belief network software developed on a windows-based platform, which makes the models not easily portable and is limited in its graphical representation across multiple system platforms. They present a prototype of BN models and influence diagrams in a World Wide Web environment, which can be displayed by a standard web browser.

Tungkasthan, et al presents a visualization of BN and influence Diagram models on a website (Tungkasthan et al., 2008). They develop an application based on the Macromedia Flash and Flash Remoting technologies. The application model on the client side is constructed by using the Macromedia Flash and the connection between a client and web server is developed by using the Flash Remoting technology. They use the capability of Marcomedia Flash and Flash Remoting technology to build richer, more interactive, more efficient, and more intuitive user interfaces for their applications than are possible with other web technologies such as JSP and Java applets. Their applications also provide a powerful, intuitive drag-and-drop graphical authoring tool that is comfortable for the users and have quick-loading and dynamic interfaces.

Jongsawat, et al presents a SMILE web-based interface that permits users to build a BN causal structure from a dataset or database and perform Bayesian network diagnosis through the web (Jongsawat & Premchaiswadi, 2009). There are several learning algorithms such as Greedy Thick Thinning, PC, Essential Graph Search, and Naive Bayes provided for the user. The user can just select the desired learning algorithm and adjust its parameter settings to learn the model structure. After building the BN structure, the user is able to quantify uncertain interactions among random variables by setting observations (evidence)

and use this quantification to determine the impact of the observations. The SMILE web-based interface was developed based on SMILE, SMILEarn, and SMILE.NET. It uses a novel, user-friendly interface which interweaves the steps in the BN analysis with brief support instructions on the web page. They also present a technique to dynamically feed data into a diagnostic BN model and a web-based user interface for the models (Jongsawat et al., 2010). In their work, the BN model (the students' attitude towards several factors in a college enrolment decision) is fixed and the data obtained from an online questionnaire are saved into a database and transferred to the model. The user can observe the changes in the probability values and the impact the changes have on each node in real-time after clicking on a belief update button. Users can also perform Bayesian inference in the model and they can compute the impact by observing values of a subset of the model variables on the probability distribution over the remaining variables based on real-time data. They also present a methodology based on group decision making for weighting expert opinions or the degree of an expert's belief in identifying the causal relationships between variables in a BN model (Jongsawat et al., 2010).

3. Tools to build a bayesian network causal structure from a dataset

The core reasoning engines of the web-based interface development capability proposed in this article consist of SMILE (Structural Modeling, Inference, and Learning Engine), SMILEarn, and JSMILE. SMILE is a reasoning engine that is used for graphical probabilistic models and provides functionality to perform diagnosis. SMILEarn is used for obtaining data from a data source, pre-processing the data, and learning the causal structure of BN models. JSMILE is used for accessing the SMILE library from the web-based interface. This section provides some more detailed information about SMILE, SMILEarn and JSMILE wrapper.

SMILE is a fully platform independent library of functions implementing graphical probabilistic and decision-theoretic models, such as Bayesian networks, influence diagrams (IDs), and structural equation models (Druzdzel, 1999). Its individual functions, defined in the SMILE Application Programmer Interface (API), allow creating, editing, saving, and loading graphical models, and using them for probabilistic reasoning and decision making under uncertainty. SMILE can be embedded in programs that use graphical probabilistic models as their reasoning engines. Models developed in SMILE can be equipped with a user interface that best suits the user of the resulting application. SMILE is written in C++ in a platform-independent manner and is fully portable. Model building and the reasoning process are under full control of the application program as the SMILE library serves merely as a set of tools and structures that facilitates them.

SMILEarn extends the functionality provided by SMILE. It provides a set of specialized classes that implement learning algorithms and other useful tools for automatically building graphical models from data. It is a C++ library that contains a set of data structures, classes, and functions that implement learning algorithms for graphical models and includes other functionality (such as data access, storage and pre-processing) that can be used in a model in conjunction with SMILE. Although SMILEarn is a module of SMILE, which means that it requires SMILE to be used, but one can use SMILE without the need to install and use SMILEarn.

JSMILE is a library of java classes for reasoning about graphical probabilistic models, such as Bayesian networks and influence diagrams. It can be embedded in programs that use graphical probabilistic models as a reasoning engine. It is a wrapper library that enables access to the SMILE and SMILEXML C++ libraries from java applications. JSMILE is not limited to stand-alone applications. It can also be used on the back-end side of a multi-tiered application.

4. Weighting expert opinions scheme

We apply the weighting expert opinions scheme to the BN model, which is constructed based on the core reasoning engines mentioned in previous section. In this section we present the sequence of steps in the decision making procedure using the weighting expert opinions scheme. The sequence of decision procedure is described as follows.

Let $V = \{v_1,…,v_m\}$ be a set of decision makers (or experts) who present their opinions on the pairs of a set of alternatives $X = \{x_1,…,x_n\}$ where m is the number of experts and n is the number of alternatives in a set. Both m and n must be greater than or equal to 3; m, n ≥ 3. $P(V)$ denotes the power set of $V(I \in P(V))$. Linear orders are binary relations satisfying reflexivity, antisymmetry and transitivity, and weak orders (or complete preorders) are complete and transitive binary relations. With $|I|$ we denote the cardinality of I.

We consider that each expert classifies the alternatives within a set of linguistic categories $L = \{l_1,…,l_q\}$, with q ≥ 2, linearly ordered $l_1 > l_2 > … > l_q$ (Herrera, 2000: Yager, 1993). The individual assignment of each expert v_i is a mapping $C_i = X \rightarrow L$ which assigns a linguistic category $C_i(x_u) \in L$ to each alternative $x_u \in X$. Associated with C_i, we consider the weak order R_i defined by $x_u R_i x_v$ if $C_i(x_u) \geq C_i(x_v)$. It is important to note that experts are not totally free in declaring preferences. They have to adjust their opinions to the set of linguistics categories, so the associated weak orders depend on the way they sort the alternatives within the fixed scheme provided by $L = \{l_1,…,l_q\}$. For instance, for q = 5 expert-1 can associate the assignment: $C_1(x_3) = l_1, C_1(x_1) = C_1(x_2) = C_1(x_4) = l_2, C_1(x_5) = l_3, C_1(x_6) = C_1(x_7) = l_4, C_1(x_8) = C_1(x_9) = l_5$; expert 2 can associate the assignment: $C_2(x_1) = l_1, C_2(x_4) = l_2, C_2(x_5) = l_3, C_2(x_7) = C_2(x_8) = l_4, C_2(x_2) = C_2(x_3) = C_2(x_6) = l_5$; and so on. A profile is a vector $C = (C_1,…,C_m)$ of individual assignments. We denote by C the set of profile.

Every linguistic category $l_k \in L$ has associated a score $s_k \in R$ in such a way that $s_1 \geq s_2 \geq … \geq s_p$. For the expert v_i, let $S_i \rightarrow R$ be the mapping which assigns the score to each alternative, $S_i(x_u) = s_k$ whenever $C_i(x_u) = l_k$. The scoring vector of v_i is $(S_i(x_1),…,S_i(x_n))$.

Naturally, if $s_i > s_j$ for all i, j ∈ {1,…,q} such that i > j, then each linguistic category is determined by its associated score. Thus, given the scoring vector of an expert we directly know the way this individual sorted the alternatives. Although linguistic categories are equivalent to decreasing sequences of scores, there exist clear differences from a behavioral point of view.

4.1 Sort the alternatives and assign a score

Experts $\{v_1,…,v_m\}$ sort the alternatives of $X = \{x_1,…,x_n\}$ according to the linguistic categories of $L = \{l_1,…,l_q\}$. Then, we obtain individual weak orders $R_1,…,R_m$ which ranks the alternatives within the fixed set of linguistic categories. Next, taking into account the scores

$s_1,\ldots,s_p$ associated with $l_1,\ldots,l_q$, a score is assigned to each alternative for every expert: $S_i(x_u)$, $I = 1,\ldots m; u = 1,\ldots,n$.

4.2 Calculate the euclidean distance

In order to have some information about the agreement in each subset of experts, we first calculate a distance between pairs of preferences (scoring vector). Since the arithmetic mean minimizes the sum of distances to individual values with respect to the Euclidean metric, it seems reasonable to use this metric for measuring the distance among scoring vectors. Let $(S(x_1),\ldots,S(x_n))$ and $(S'(x_1),\ldots,S'(x_n))$ be two individual or collective scoring vectors. The distance between these vectors by means of the Euclidean metric is derived by (1).

$$d(s,s') = \sqrt{\sum_{u=1}^{n} (s(x_u) - S'(x_u))^2} \tag{1}$$

4.3 Aggregate the expert opinions

We aggregate the expert opinions by means of collective scores which are defined as the average of the individual scores. There are several steps in this procedure.

4.3.1 Calculate the overall agreement measure

We calculate a specific agreement measure which is based on the distances among individual and collective scoring vectors in each subset of experts. The overall agreement measure is derived by (2).

$$M(C_r I) = 1 - \frac{\sum_{V_i \in I} d(S_i, S)}{|I| S_1 \sqrt{n}} \tag{2}$$

We note that $S_1 \sqrt{n}$ is the maximum distance among scoring vectors, clearly between $(S(x_1),\ldots, S(x_n)) = (s_1,\ldots, s_1)$ and $(S'(x_1),\ldots, S'(x_n)) = (0,\ldots,0)$; $d(S, S') = S_1\sqrt{n}$. $M(C, I)$ is equal to 0 if $I = \phi$. Then, $M(C, I) \in [0, 1]$, for every $(C, I) \in C \times P(V)$. It is easy to see that the overall agreement measure satisfies the other axioms of (Bosch, 2005), Anonymity and Neutrality.

4.3.2 Calculate the overall contribution to the agreement

We now calculate an index which measures the overall contribution to agreement by each expert with respect to a fixed profile, by adding up the marginal contributions to the agreement in all subsets of experts. The overall contribution to the agreement of expert v_i with respect to a profile is defined by (3).

$$W_i = \sum_{I \subset V} M(C,I) - M(C,I \setminus |v_i|)) \tag{3}$$

If $w_i > 0$, we can conclude that expert v_i positively contributes to the agreement; and if $w_i < 0$, we can conclude that that expert v_i negatively contributes to the agreement.

4.3.3 Calculate the weak order

We now introduce a new collective preference by weighting the score which experts indirectly assign to alternatives with the corresponding overall contribution to the agreement indices. The collective weak order associated with the weighting vector w = $(w_1,…,w_m)$, R^w, is defined by (4) and (5).

$$W_i = \sum_{I \subset V} (M(C,I) - M(C,I \setminus |v_i|)) \qquad (4)$$

where

$$S^w(x_u) = \frac{1}{m}\sum_{i=1}^{m} w_i S_i(x_u) \qquad (5)$$

Consequently, we prioritize the experts in order of their contribution to agreement (Cook et al., 1996).

5. SMILEBN web application

The following steps in this section describe how a SMILEBN web application works for creating the BN models based on the combination of structure learning algorithms and weighting expert opinions scheme. The structure of the proposed framework is presented in Fig. 1. It shows a practical framework for building diagnostic Bayesian networks based on both learning algorithms and expert beliefs.
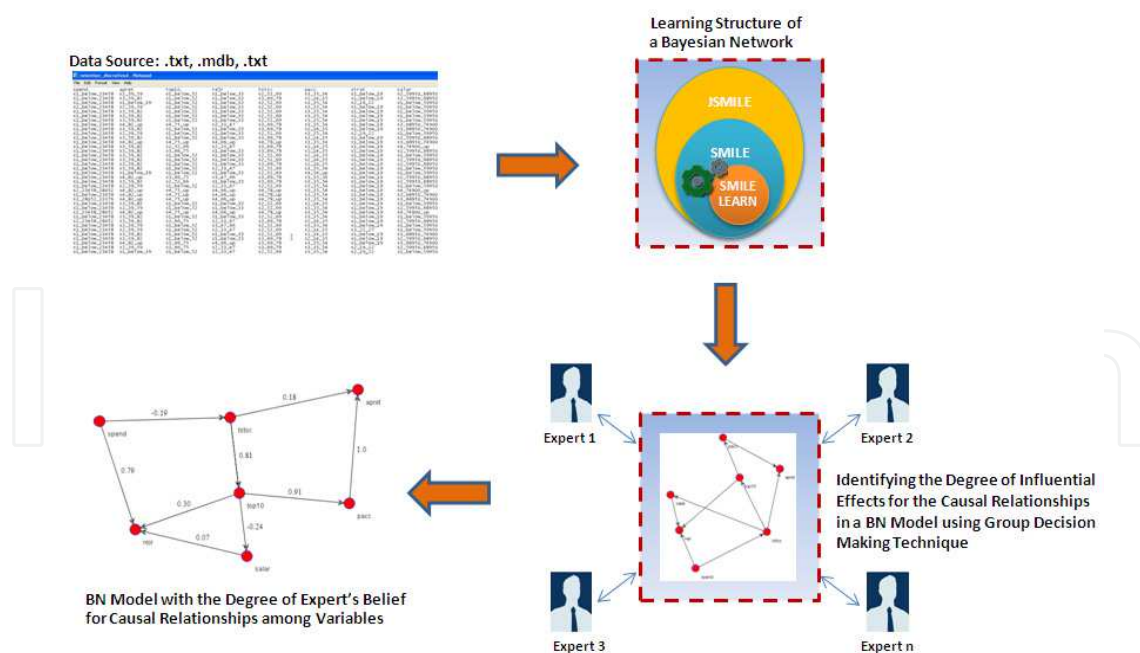
Fig. 1. A Practical framework for building diagnostic Bayesian networks based on both learning algorithms and expert beliefs

The first step is to import the data from a database or the data stored in the text file to the SMILEBN web application. Users select the file from the list and then clicks on "OK" button.

SMILEBN uses the data grid view to display the loaded data files and let's users work with them much like with spreadsheets. If the data file does not contain any missing values, SMILEBN will inform the users about that and "Next" button will be enabled. Otherwise, SMILEBN will tell how many rows were selected and the corresponding ones will become highlighted in the data grid. Users must solve the missing values manually (See Fig. 2 and Fig. 3). Once they have a data set prepared they can proceed to learning the network by picking the method and setting it's parameters. Note that if the data set contains continuous variables they will need to be discretized for some learning methods to be able to run, e.g. Naive (See Fig. 4 and Fig. 5). Fig. 6 shows the structure of a Bayesian network after applying the learning process. It shows the probability values over all nodes after performing Bayesian updating or belief updating (by clicking on "Update Belief" button) when the users move the mouse cursor over any node (See Fig.7). The user is allowed to perform a model diagnosis by entering observations (evidence) for some of the context and evidence variables. Fig. 8 shows the screenshot of the BN model diagnosis. The user begins the BN model diagnosis by performing a right click on a node and selects the state for setting the evidence for the test.

After setting the evidence, they click on the "Update Belief" button to update the model. Fig. 2 - Fig. 8 mainly show the methods to build a BN model in the SMILEBN web application based on the structure learning algorithms mentioned in section 3. Next the weighting expert opinions scheme will be applied to the BN model.
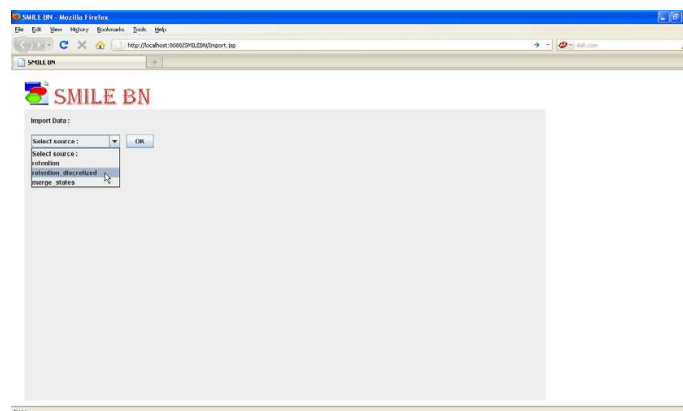
Fig. 2. Importing the data files
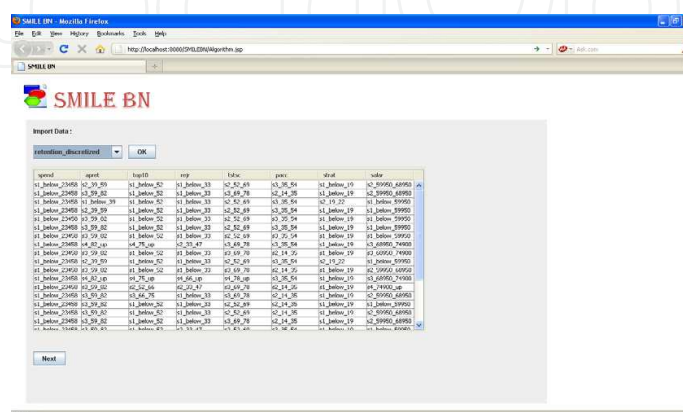
Fig. 3. Data grid view to display the loaded data files
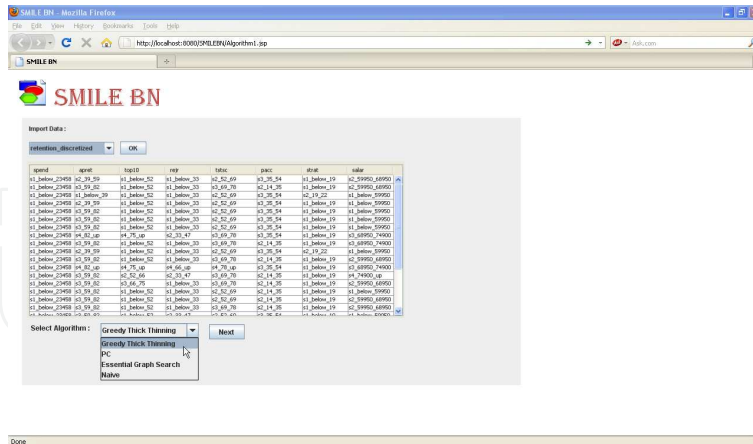
Fig. 4. Selecting the learning algorithms
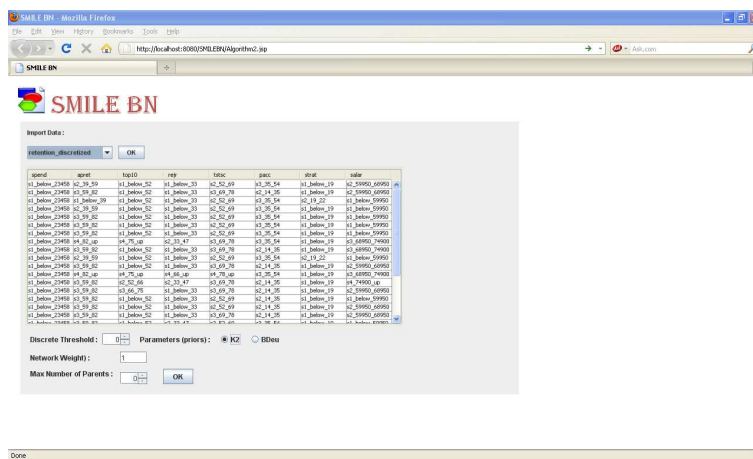
Fig. 5. Setting learning algorithm's parameter
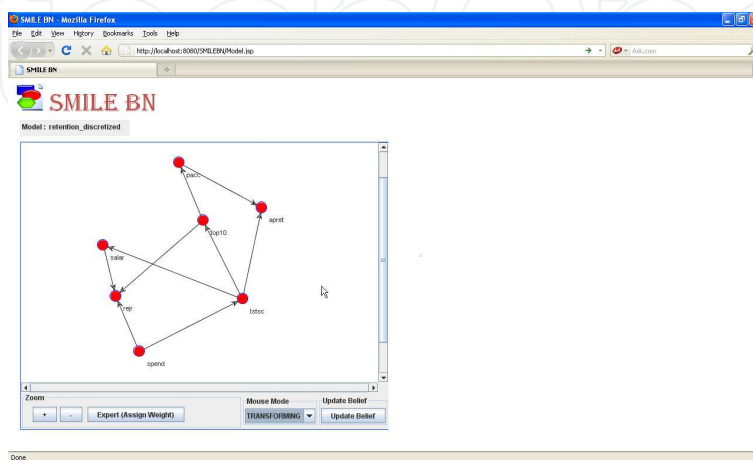
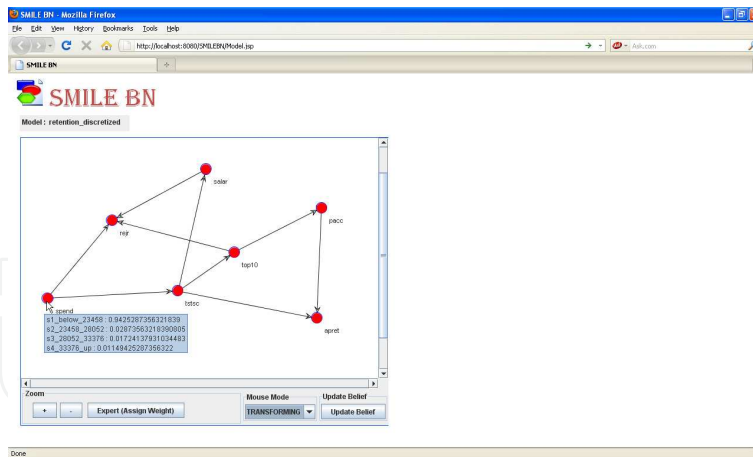Fig. 6. The structure of a Bayesian network after applying the learning process

Fig. 7. Bayesian network with the probability values over all nodes after performing Bayesian updating (belief updating)
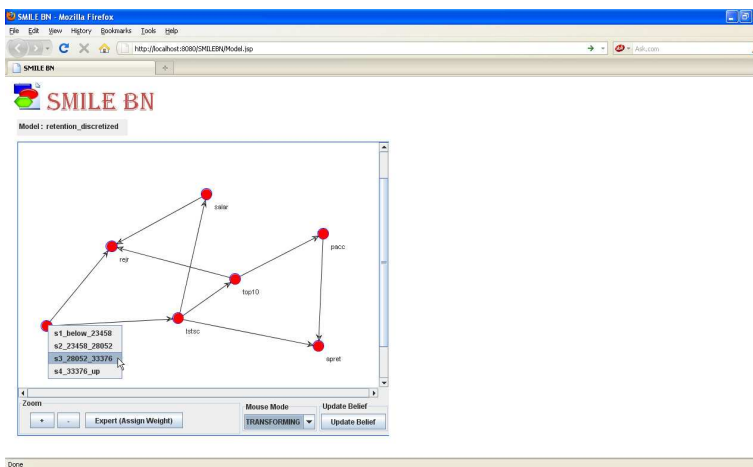

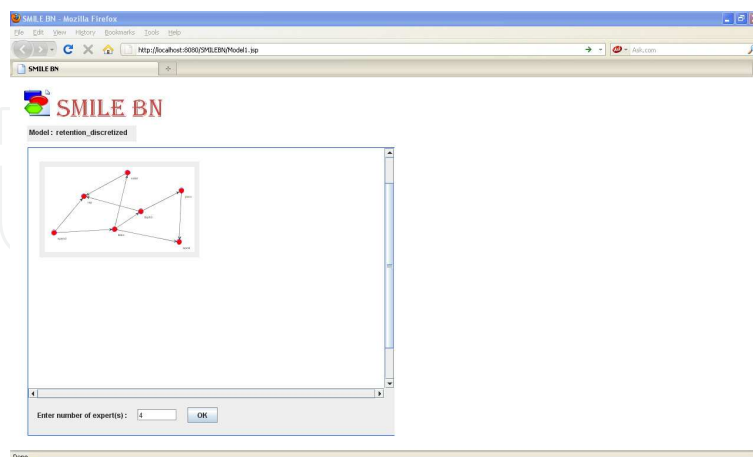Fig. 8. Selecting an evidence corresponding to the node

Fig. 9. Specifying the number of the expert(s)

We propose a methodology based on group decision making for weighting expert opinions or the degree of an expert's belief in identifying the causal relationships between variables in a BN model. The idea is to find the final BN solution that is obtained from a group of

experts and to minimize the number of relationships among the nodes in the model for simplicity by setting a threshold value. The methodology consists of three sequential steps.

First, in a pre-processing step, all the experts in group must agree with each other for the BN model that is built based on the structure learning algorithms.

Second, we map every pair of causal variables into alternatives. Then, experts sort the alternatives by means of a fixed set of linguistic categories; each one has associated a numerical score. We average the scores obtained by each alternative and we consider the associated preference. Then we obtain a distance between each individual preference and the collective one through the Euclidean distance among the individual and collective scoring vectors. Taking into account these distances, we measure the agreement in each subset of experts, and a weight is assigned to each expert. We calculate the collective scores after we weight the opinions of the experts with the overall contributions to agreement. Those experts whose overall contribution to the agreement is negative are excluded and we re-calculate the decision procedure with only the opinions of the experts which positively contribute to agreement. The sequential decision procedure is repeated until it determines a final subset of experts where all of them positively contribute to agreement for group decision making. Lastly, we transform the alternatives and the collective scores that we obtain from previous step into the BN models. The mathematical formulas for this scheme are mentioned in section 4.

In the application point of view, users select the number of the experts (See Fig.9). In this example, we have a group of four experts who participate in identifying the degree of influential effects for the causal relationships in a BN model. The level of influential effects among the nodes based on each expert's belief is specified (See Fig.10). Each expert is asked to perform this task one by one. When all experts have completed this task, the BN model with the degree of expert's belief among causal relationship variables in the initial step of the decision procedure is presented (See Fig.11). Fig.12 shows the BN model and the degree of expert's belief among variables in normalized form (0..1) when users click on the "Normalized" button. Fig.13 shows the simplified BN model in the initial step of the decision procedure when users set a threshold value and click on "OK" button. They can select the other steps of the decision procedure from the list in a combo box below the model window and perform the same steps as presented in Fig.12 and Fig.13. The number of steps of the decision procedure depends on the number of expert and the ways they identify the degree of influential effects for the causal relationships in a BN model. Fig.14 – Fig.16 shows the BN
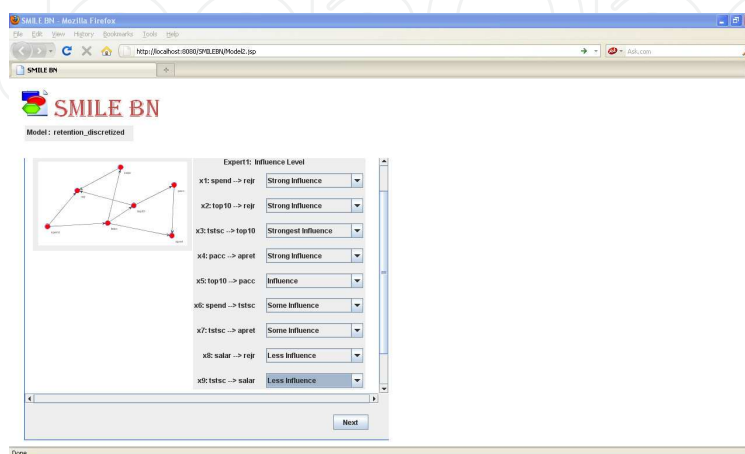

Fig. 10. Specifying the level of influential effects among the nodes based on expert's belief

model, the model in normalized form, and the model with a threshold value = 0.2 in the first step of the decision procedure. Fig.17 – Fig.19 shows the BN model, the model in normalized form, and the model with a threshold value = 0.3 in the second step of the decision procedure.
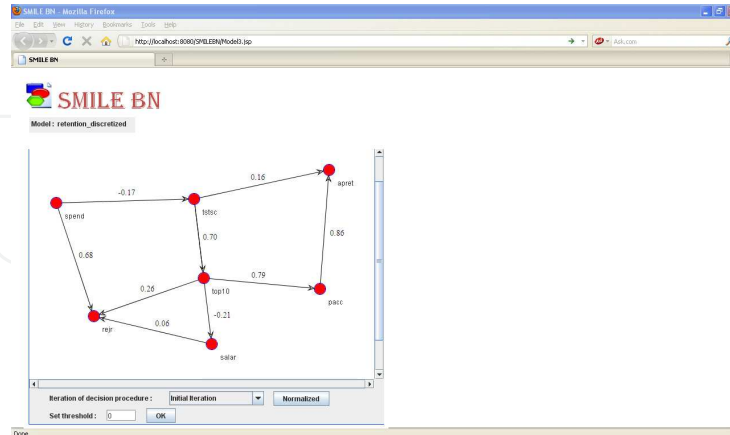

Fig. 11. BN model and the degree of expert's belief among causal relationship variables (initial step of the decision procedure)
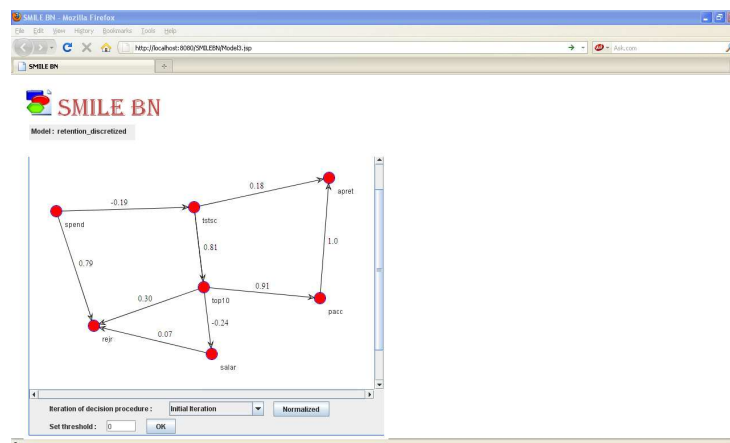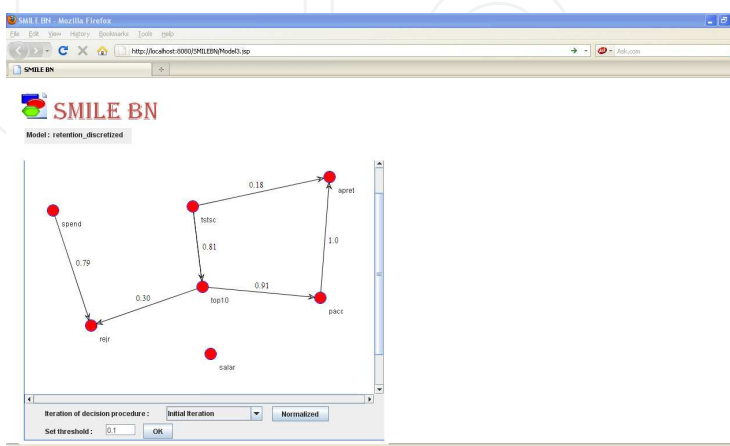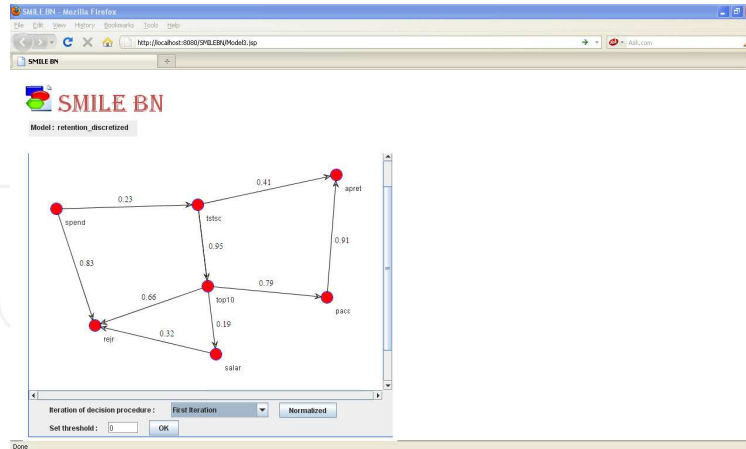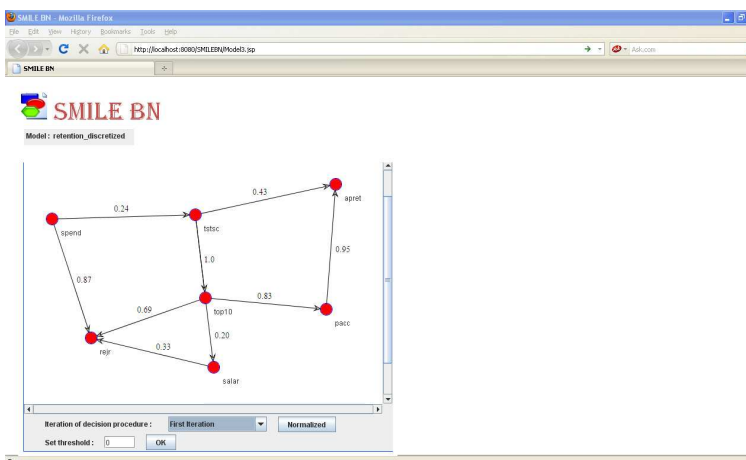

Fig. 12. BN model and the degree of expert's belief among causal relationship variables in normalized form (initial step of the decision procedure)

Fig. 13. BN model and the degree of expert's belief among causal relationship variables after applying the threshold value (initial step of the decision procedure)

Fig. 14. BN model and the degree of expert's belief among causal relationship variables (first iteration of the decision procedure)

Fig. 15. BN model and the degree of expert's belief among causal relationship variables in normalized form (first iteration of the decision procedure)
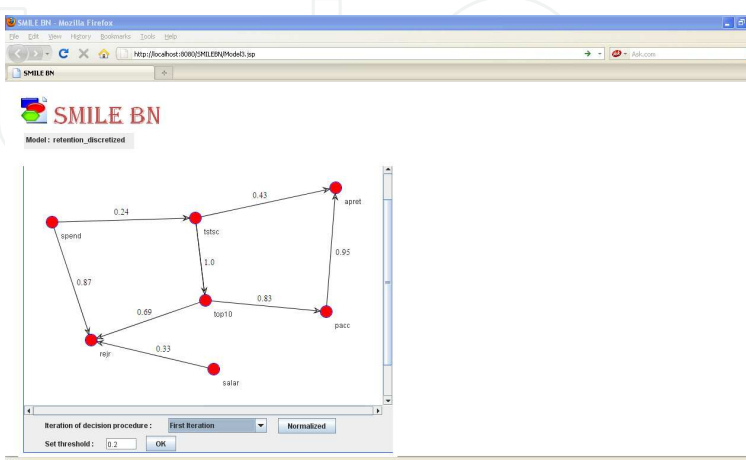

Fig. 16. BN model and the degree of expert's belief among causal relationship variables after applying the threshold value = 0.2 (first iteration of the decision procedure)
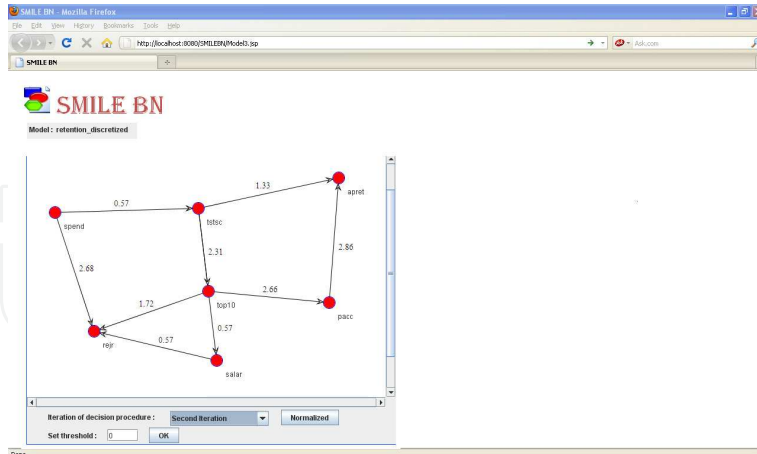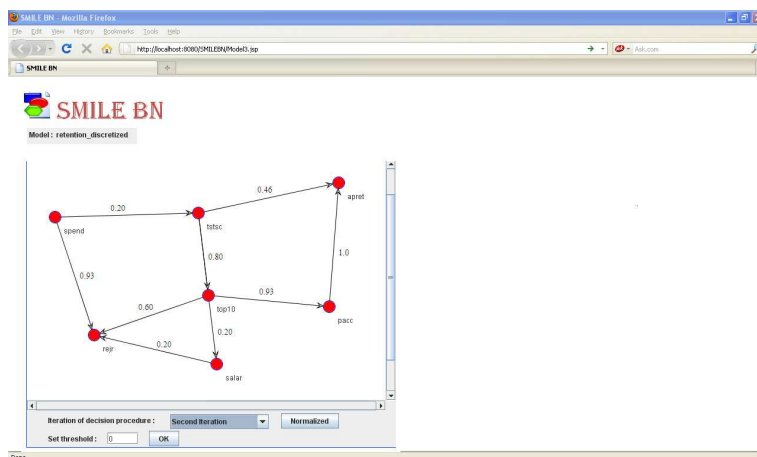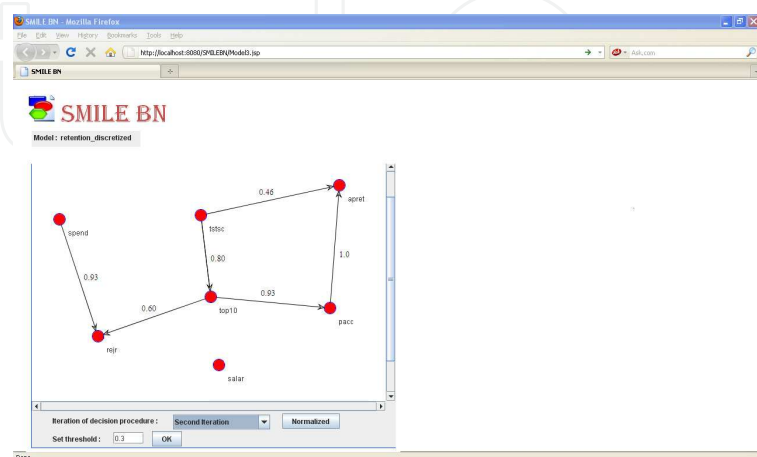
Fig. 17. BN model and the degree of expert's belief among causal relationship variables
(second iteration of the decision procedure)

Fig. 18. BN model and the degree of expert's belief among causal relationship variables in
normalized form (second iteration of the decision procedure)

Fig. 19. BN model and the degree of expert's belief among causal relationship variables after
applying the threshold value = 0.3 (first iteration of the decision procedure)

6. Conclusion and future work

This article presents a SMILEBN web application for building a Bayesian network model. The SMILEBN can build a BN model based on using two approaches. First, a BN model is built by applying the structure learning algorithms to a dataset. The variables in a dataset can be both discrete and continuous variables. The core reasoning engines of the SMILEBN web application consist of SMILE, SMILEarn, and JSMILE. SMILE is used for graphical probabilistic models and provides functionality to perform diagnosis. SMILEarn is used for obtaining data from a data source, pre-processing the data, and learning the causal structure of BN models. JSMILE is used for accessing the SMILE library from the web-based interface. Second, group decision making technique for weighting expert opinions scheme is applied to the BN model. This scheme is used to identify influential effects from parent variables to child variables in the BN model based on having information about the agreement and overall agreement measure produced by a group of experts. The sequential decision procedure is repeated until it determines a final subset of experts where all of them positively contribute to agreement for group decision making. Several steps of the decision procedure will be generated. The aims of the second approach are that we need to obtain the BN model, which all the experts agree to use, and to minimize the number of relationships among the nodes in the model for simplicity by setting a threshold value. When the number of relationships among the nodes decreases, the complexity of the conditional probability table on each child node also decreases.

Our future work will focus on improving a decision-oriented diagnosis approach. The SMILEBN will be extended to cope with influence or relevance diagrams.

7. Acknowledgement

8. References

Bosch, R. (2005). Characterizations of voging rules and consensus measures. *Ph.D. Dissertation, Tilburg University.*

Cook, W. D.; Kress, M. & Seiford L. M. (1996). A general framework for distance-basedconsensus in ordinal ranking models. *European Journal of Operational Research*, pp. 392–397.

Druzdzel, M.J. (1999). SMILE: Structural Modeling, Inference, and Learning Engine and GeNIe: A Development Environment for Graphical Decision-Theoretic Models. *In Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI–99)*, p. 902-903, Orlando, FL.

Herrera, F. & Herrera-Viedma, E. (2000). Linguistic Decision Analysis: Steps for Solving Decision Problems under Linguistic Information. *Fuzzy Sets and Systems 115*, pp. 67–82.

Jongsawat, N. & Premchaiswadi, W. (2009). A SMILE Web-Based Interface for Learning the Causal Structure and Performing a Diagnosis of a Bayesian Network. *Proceedings of IEEE on Systems, Man, and Cybernetic, Systems Science & Engineering (SMC2009)*, San Antonio, Texas, USA.

Jongsawat, N. & Premchaiswadi, W. (2010). Weighting Expert Opinions in Group Decision Making for the Influential Effects between Variables in a Bayesian Network Model. *Proceedings of IEEE on Systems, Man, and Cybernetic, Systems Science & Engineering (SMC2010)*, Istanbul, Turkey, pp.1029-1035.

Jongsawat, N.; Poompuang, P. & Premchaiswadi, W. (2010). Dynamic Data Feed to Bayesian Network Model and SMILE Web Application. *Bayesian Network edited by Dr.Ahmed Rabai*, p. 155-166.

Lacave, C.; Luque, M. and Díez, F.J. (2007). Explanation of Bayesian Networks and Influence Diagrams in Elvira. *IEEE Transactions on Systems, Man, AND Cybernetics-Part B: Cybernetics*, Vol. 37, No. 4, p. 952-965.

Mylltmaki, P.; Silander, T.; Tirri, H. & Uronen, P. (2002). B-Course a Web-based Tool for Bayesian and Causal Data Analysis. *International Journal on Artificial Intelligence Tools*, Vol. 11, No. 3, p. 369-387.

Poompuang, P.; Kungtasthan, A. ; Jongsawat, N. & Sutheebanjard, P. (2007). Graphical Decision-Theoretic Models on the Web. *Proceedings of Knowledge Management*, p. 163-170, Bangkok, Thailand.

Premchaiswadi, W. & Jongsawat, N. (2010). Bayesian Network Inference with Qualitative Expert Knowledge for Group Decision Making. *Proceedings of IEEE on on Intelligent Systems (IS 2010)*, London, UK, pp.126-131.

Tungkasthan, A.; Poompuang, P. & Premchaiswadi, W. (2008). SMILE Visualization with Flash Technologies, *Proceedings of IEEE on Software Engineering: Artificial Intelligence, Networking, and Parallel/Distributed Computing*, p. 551-556, Phuket, Bangkok, Thailand.

Yager, R. R. (1993). Non-Numeric Multi-Criteria Multi-Person Decision Making. *Journal of Group Decision and Negotiation 2*, pp. 81–93.
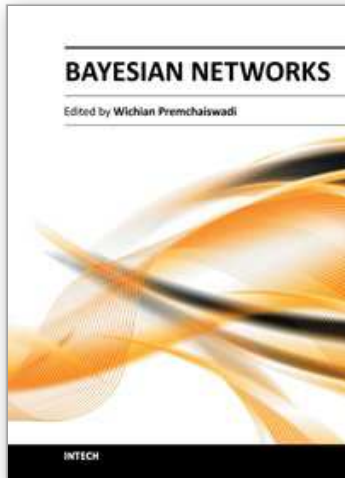
Bayesian Networks

Edited by Dr. Wichian Premchaiswadi

Bayesian Belief Networks are a powerful tool for combining different knowledge sources with various degrees of uncertainty in a mathematically sound and computationally efficient way. A Bayesian network is a graphical model that encodes probabilistic relationships among variables of interest. When used in conjunction with statistical techniques, the graphical model has several advantages for data modeling. First, because the model encodes dependencies among all variables, it readily handles situations where some data entries are missing. Second, a Bayesian network can be used to learn causal relationships, and hence can be used to gain an understanding about a problem domain and to predict the consequences of intervention. Third, because the model has both causal and probabilistic semantics, it is an ideal representation for combining prior knowledge (which often comes in a causal form) and data. Fourth, Bayesian statistical methods in conjunction with Bayesian networks offer an efficient and principled approach to avoid the over fitting of data.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Wichian Premchaiswadi and Nipat Jongsawat (2012). Building a Bayesian Network Model Based on the Combination of Structure Learning Algorithms and Weighting Expert Opinions Scheme, Bayesian Networks, Dr. Wichian Premchaiswadi (Ed.), ISBN: 978-953-51-0556-5, InTech, Available from: http://www.intechopen.com/books/bayesian-networks/building-a-bayesian-network-model-based-on-the-combination-of-structure-learning-algorithms-and-weig

INTECH
open science | open minds