

# A SMILE Web-based Interface for Learning the Causal Structure and Performing a Diagnosis of a Bayesian Network

Nipat Jongsawat

Graduate School of Information Technology in Business  
Siam University  
Bangkok, Thailand  
nipatj@yahoo.com

Wichian Premchaiswadi

Graduate School of Information Technology in Business  
Siam University  
Bangkok, Thailand  
wichian@siam.edu

**Abstract**— Bayesian networks (BNs) are probabilistic graphical models that are widely used for building diagnosis- and decision-support expert systems. The construction of 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. Learning the structure of a Bayesian network model and causal relations from a dataset or database is important for large BNs analysis. This paper focuses on using a SMILE web-based interface for building the structure of BN models from a dataset by using different structural learning algorithms. In addition to building the structure of BN models, a SMILE web-based interface also provides the feature set of Bayesian diagnosis for the user. The web application uses a novel user-friendly interface which intertwines the steps in the data analysis with brief support instructions to the Bayesian approach adopted. A SMILE web-based interface has been developed based on SMILE (Structural Modeling, Interface, and Learning Engine), SMILEarn, and SMILE.NET wrapper.

**Keywords**— Bayesian network, Bayesian diagnosis, decision-support expert systems, update belief, SMILE, SMILEarn, SMILE.NET wrapper, SMILE web-based interface

## I. INTRODUCTION

A Bayesian network is a graphical probabilistic model that is widely used for building expert systems in several application domains. It consists of acyclic directed graphs (ADGs) and probability distributions [1] – [3]. It represents a set of variables and their probabilistic independencies. Formally, Bayesian networks are ADGs in which nodes represent random variables and arcs represent direct probabilistic dependencies among them. A Bayesian network also represents the quantitative relationships among the modeled variables. Numerically, it represents a joint probability distribution. This distribution is efficiently described by exploring the probabilistic independencies among the model variables. Each node is described by a probability distribution conditional on its direct predecessors. Nodes with no predecessors are described by prior probability distributions [4]. The causal structure and the numerical parameters of a Bayesian network can be obtained using two distinct approaches [5]. First, they can be obtained from an expert. Second, they can also be learned from a dataset or database because the structure of a Bayesian network is simply a

representation of independencies in the data and the numerical values are a representation of the joint probability distributions that can be inferred from the data [17, 18]. In practice, some combination of these two approaches is typically used. For example, the causal structure of a model is acquired from an expert, while the numerical parameters of the model are learned from a database.

However, this paper focuses primarily on learning causal Bayesian network structures from data or a database in the WWW environment. It is a very difficult task for experts to identify the correct causal structure and the numerical parameters of Bayesian network models when the problem is very complicated. Large BN models consist of many variables [20]. Different experts often think about the same problem in quite different ways. Thus, the most suitable way for learning the causal structure and the numerical parameters of a Bayesian network model when a complete dataset has already been prepared is to learn the structure by using various well-known BN structure learning algorithms such as Greedy Thick Thinning, PC, Essential Graph Search or Naive Bayes. After obtaining the causal structure, the experts or even inexperienced users are usually able to use their own judgment to critically evaluate the modeling and the inference results. Furthermore, they can perform Bayesian diagnosis on a probabilistic model by adjusting the model to a new situation. This allows them to query the system subsequently about the new, posterior probability distributions. Most of the work presented in the literature focuses on using BN software for building a dependency model and performing diagnosis of different aspects. This paper presents a SMILE web-based interface for learning the structure of a Bayesian network and performing diagnosis on the model. The reason behind this approach is that working in the World Wide Web environment is more flexible for analysis and diagnosis reasoning. Experts, modelers, or users can participate in web-based collaborative work for discussions about the model without any additional software installation. They can also share their model and ideas easily using the web. The web-based interface presented in this paper uses the following engines, namely, SMILE (Structural Modeling, Interface, and Learning Engine), SMILEarn, and SMILE.NET which is used as a wrapper by calling the

functions, passing the parameter values to these engines and receiving the return values.

Each section of this paper is organized as follows: Section II describes the core of the web-based interface development: Structural Modeling, Inference, and Learning Engine (SMILE), SMILEarn, and SMILE.NET wrapper. Section III describes the related work. Section IV describes the implementation and the use of a web-based interface for learning the causal structure and performing diagnosis of a Bayesian network in more detail. Section V presents conclusion and gives some perspectives and ideas for future work.

## II. BACKGROUND

The core engines of the web-based interface development consist of SMILE (Structural Modeling, Inference, and Learning Engine), SMILEarn, and SMILE.NET. 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, preprocessing the data, and learning the causal structure of BN models. SMILE.NET is used for accessing the SMILE library from the web-based interface. This section provides some more detailed information about SMILE, SMILEarn and SMILE.NET wrapper.

### A. SMILE

SMILE [6, 7] is a fully platform independent library of C++ classes implementing graphical probabilistic and decision-theoretic models, such as Bayesian networks, influence diagrams, and structural equation models. Its individual classes, defined in the SMILE Application Programming Interface (API), allow the user to create, edit, save, and load graphical models, and use them for probabilistic reasoning and decision making under uncertainty. These classes are accessible from C++ or (as functions) from the C programming language. SMILE can be embedded in programs that use graphical probabilistic models as their reasoning engines. Furthermore, models developed in SMILE can be equipped with a GUI that suits the user of the resulting application most appropriately.

The diagnostic part of SMILE provides functionality to perform diagnosis on any network in SMILE. With SMILE diagnosis a user can determine a state of the network by performing tests or observations. A user is able to select a test and perform it by setting the evidence for the test. There are several important classes and functions in SMILE. For example, the functions for assigning the nodes and states are provided by the class `DSL_extraDefinition`. The functions to perform diagnosis are provided by the class `DIAG_network`. To determine the usefulness or value of a test, the class `DSL_valueOfInformation` is used to determine the expected value of observing a test. For this paper, SMILE is mainly used for setting observations, updating belief (diagnosis) and building graphical probabilistic models.

### B. SMILEarn

SMILEarn [8] 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 preprocessing) that can be used in model 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.

SMILEarn divides the process of learning graphical models into several conceptual steps. In the most general case, learning consists of the following three steps: (1) obtaining raw data from a data source, (2) preprocessing the data, and (3) applying a learning procedure that results in a model (or some equivalent structure). The raw data in step 1 can be a text file, an external database or a set of observations associated with a model. The preprocessing step is an optional step that amounts to removing or filling missing values in the dataset, discretization of continuous values, removing outliers, etc. Step 3 applies a learning algorithm to the preprocessed data. There are several important classes in SMILEarn. For example, the class `DSL_dataset` is used for passing data between data sources and the preprocessing and learning classes. The class `DSL_textParser` is a utility class that implements a simple parser for text files. The class `DSL_discretizer` offers discretization methods that can be applied to the datasets. Besides these fundamental classes, there are several important classes for learning the causal structure of Bayesian Networks such as `DSL_naiveBayes`, `DSL_greedy`, `ThickThinning`, `DSL_pc`, and `DSL_em`.

This paper uses SMILEarn, which is a module of SMILE, for obtaining raw data from a data source, preprocessing the data and handling missing values in the dataset, as well as learning and building graphical models from data by using the various algorithms provided.

### C. SMILE.NET Wrapper

SMILE.NET [9, 19] is a library of .NET 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. SMILE.NET is not limited to stand-alone applications. It can also be used on the back-end side of a multi-tiered application.

SMILE.NET is a wrapper library that enables access to the SMILE and SMILEXML C++ libraries from .NET applications. SMILE.NET is not functionally identical to SMILE/SMILEXML in that it does not provide access to all the properties and methods presented in the original C++ libraries. However, the essential parts of the SMILE/SMILEXML interface have been exposed. SMILE.NET is easier and more intuitive to use than the two original C++ libraries. It is possible to tailor the library according to any special needs, since its source code is provided.

The SMILE web-based interface presented in this paper is developed in ASP.NET and C#.NET so that the SMILE.NET wrapper was chosen for accessing the SMILE and SMILEarn.

### III. RELATED WORK

There are various kinds of software packages 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 packages available. The commercial software 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 is extensively used for the educational purposes. These software packages have been used in teaching courses and performing research. Some of them provide open source code and allow users to fully develop and implement the code for many different purposes. Since there are a number of software packages based on BNs, this paper reviews only the most relevant subset of them. The specific software packages or programs included in this section have the following properties: (1) non-commercial software programs, (2) uses a graphical click-and-drop interface, (3) concerned with the creation of decision theoretic models, (4) have the ability to learn the causal structure of BNs from a dataset and/or performing diagnosis on both web- and non web-based applications.

B-Course [11] is an analysis tool that was developed in the fields of Bayesian and causal modeling. B-Course is a free web-based online data analysis tool, which allows users to analyze data for multivariate probabilistic dependencies. B-Course 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). B-Course offers a simple three step procedure (data upload, model search, 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 with 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, modeling assumptions, restrictions, model search algorithms, and parameter settings are totally transparent to the user.

GeNIe (Graphical Network Interface) [10] is a versatile and user friendly development environment for building graphical decision models. The original interface was designed for SMILE which is described in a previous section. 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, the GeNIe software must be installed. In order to use GeNIe efficiently, the user should have some background knowledge about probabilistic graphical models and become familiar with the tools provided in GeNIe.

Elvira [12, 13] is a tool for building and evaluating graphical probabilistic models. 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 in 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 and a database, (3) processing - implementing the algorithms for processing and evaluating models, and (4) visualization - defining the Elvira graphical user interface (GUI) and, obviously, makes use of the classes that are included in the previous modules.

J. Nipat et al [14] presents a technique to dynamically feed data into a diagnostic Bayesian network model and a web based user interface for the models. In their work, the BN model (the students' attitude towards several factors in a college enrollment decision) is fixed and the data obtained from the online questionnaire are saved into the 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 of observing values of a subset of the model variables on the probability distribution over the remaining variables based on real-time data.

The applications mentioned above have been used mainly for educational purposes. They work very well in their environment according to their goals and objectives. Some applications must be installed and some of them are web-based application. Each has its pros and cons. Some of them do not provide important features such as learning structure algorithms, obtaining the data from a data source in real time, or collaborative work for analysis. The lack of these features may be considered limitations of its usefulness. This paper attempts to incorporate these limitations and develop their associated features into a SMILE web-based interface. One of the attribute of this approach is that a SMILE web-based interface can be used on a worldwide basis.

### IV. IMPLEMENTATION OF THE WEB-BASED INTERFACE FOR LEARNING THE CAUSAL STRUCTURE AND PERFORMING DIAGNOSIS OF A BAYESIAN NETWORK

This section describes the implementation and the use of a SMILE web-based interface for learning the causal structure and performing a diagnosis of a Bayesian network in more detail.

#### A. Layers of reasoning engine for graphical probabilistic model.

SMILE is the causal reasoning engine for a SMILE web-based interface. It is a platform independent library of C++ classes for developing graphical models. In the implementation, the "using Smile;" statement is defined as using a directive at the beginning of a program for a class. SMILEarn provides a set of specialized classes that implement

learning algorithms and other useful tools for automatically building graphical models from data. It is actually a module of SMILE. SMILE.NET is a wrapper library that enables access to the SMILE and SMILEXML C++ libraries from .NET applications. Since the SMILE web-based interface is developed in ASP.NET and C#.NET, therefore it requires SMILE.NET as its modeling library. In order to use the SMILE.NET wrapper, the smilenet.dll file must be included into the project environment (see figure 1).



Figure 1. The layers of reasoning engine for graphical probabilistic models.

#### B. Example of the class and methods used for learning the structure and parameters of the Bayesian network

This section provides as example and a brief explanation of the some important classes and methods of SMILE and SMILEarn that are used by the SMILE web-based interface for learning the BN structure.

```
protected void TextParsing(object sender, EventArgs e)
{
    DSL_textParser parser;
    parser.SetUseHeader(true);
    parser.SetTypesSpecified(true);

    if (parser.Parse("selection.mdb")!=DSL_OKAY)
        Response.Write("Parsing failed!");

    DSL_dataset d = parser.GetDataset();
    PrintDataset(d);
}
```

When the file is uploaded to the server, the data are required to be parsed before building the causal structure. The source code shown above is the method that performs the text parsing procedure. The first step is to create DSL\_textParser object and set appropriate parameters. The DSL\_textParser parameter "SetUseHeader()" is set to true (boolean) in order to read the first row in the text or database file that contains the labels of the variables. The SetTypesSpecified() specifies the header of the data types. If the parameter value of SetTypesSpecified() is true, it means that the data is discrete. The Parse() method performs actual file ("selection.mdb") parsing and creates internal data structures. Every time this method is called, the file parsing takes place. It returns DSL\_OKAY if the operation was successful, and an error code in case of an error. Finally, we create a data set object "d" that contains parsed data. This method, GetDataset(), allows obtaining a data set from parsed file.

```
class DSL_network;
class DSL_dataset;
class DSL_greedyThickThinning
{
public:
    DSL_greedyThickThinning()
    {
        maxParents = 5;
        priors = K2;
        netWeight = 1.0;
    }

    int Learn(const DSL_dataset &data, DSL_network &net);
    enum PriorsType { K2, BDeu };
    PriorsType priors;
    int maxParents;
    double netWeight;
    typedef std::vector<std::pair<int, int> > IntPairVector;
    IntPairVector forcedArcs;
    IntPairVector forbiddenArcs;
    IntPairVector tiers;
};
```

The source code shown above is the class that implements the greedy thick thinning procedure for learning the structure and parameters of a Bayesian network. There are several fields of this class that are used for defining some details of the learning algorithm. For example, the "maxParents" field defines the maximal number of parents a node can have. The "priors" field defines the type of priors (K2 method) to use [15, 16]. The learning method, Learn (const DSL\_dataset &data, DSL\_network &net), performs the actual learning procedure. The first argument is the input dataset. The result from the learning procedure is stored in the DSL\_network which is the second argument. The method returns DSL\_OKAY if learning was successful and an error code otherwise. In order to implement the learning structure in the SMILE web-based interface, the header files, smilearn.h and smile.h must be included as the header files in the program. The SMILE web-based interface calls the DSL\_greedyThickThinning() constructor that initializes the object with default values for parameters used for learning. It also calls the learning method, Learn(const DSL\_dataset &data, DSL\_network &net), for passing dataset file to perform the learning procedure. The result is stored in the DSL\_network which can be accessed by using the ReadFile() method.

#### C. A SMILE web-based interface

The SMILE web-based interface offers a simple six step procedure: (1) uploading data, (2) handling of missing values, (3) selecting variables and learning algorithms, (4) setting parameter values for the selected algorithm, (5) updating belief, and (6) performing BN diagnosis for learning and building the causal structure and performing diagnosis of a Bayesian network. The details of each step are described below.

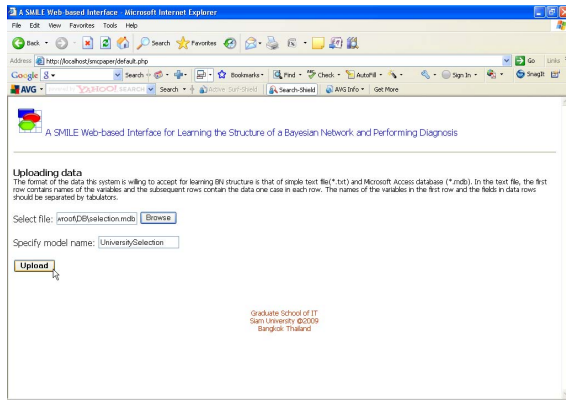


Figure 2. Screenshot of data upload.

First, uploading the data (see figure 2), the user browses for the file (.txt or .mdb) on the computer, specifies the model name, and then clicks upload. The data file is sent to the server. A SMILE web-based interface also notifies the user of possible problems such as an incompatible file format or over file size limit during the data upload. The data file used for presentation in this paper was taken from our previous work [14]. The data file was stored in a Microsoft Access database (.mdb) format that consisted of 15 columns (variables) with 400 records. The data was taken from the online questionnaire used to study students' attitudes on fifteen factors in making an enrollment decision. Four hundred students from both public and private universities were asked to fill in the online questionnaire and the data were saved into the MS Access database.

Second, handling of missing values (see figure 3), the SMILE web-based interface uses the data grid view to display the loaded data files. If the data file contains any missing values, the user must click on the handling "missing values" button to replace them. After clicking on the button, the users will be given a choice to replace the missing value (1) with a specific value or (2) with an average of the selected column. If the user selects number 1, they have to fill all blanks with a specific value manually. If the user selects number 2, the program will fill in an average of the selected column containing the blank value. When all missing values are replaced, a SMILE web-based interface will show all variables and learning algorithms on the web page.

Third, selecting variables and learning algorithms (see figure 4), the SMILE web-based interface presents simple descriptive statistics of each variable so that the user can verify that the upload was successful and the dataset was complete. As the default, all variables are selected for the learning algorithm. The user can exclude certain variables by unchecking the associated checkbox if he or she doesn't want them to be learned or part of the model. After that, the user must select an algorithm for learning the structure and click ok to continue.

Forth, setting parameters for the selected algorithm (see figure 5), once the user has a dataset prepared he or she can proceed to choosing a learning algorithm for the network. The user must specify the parameter values of the selected algorithm.

Fifth, updating belief, the user clicks on the "update belief" button for updating the probability distributions among the variables. The user is taken to a web page showing the model (see figure 6).

Finally, performing BN diagnosis, the user is allowed to perform a model diagnosis by entering observations (evidence) for some of the context and evidence variables (see figure 7). Entering observations amounts to adjusting the model to a new situation. After setting evidence values, the user clicks on the update belief button again to see the changes in the model and the impact that occurs in each variable and its context.

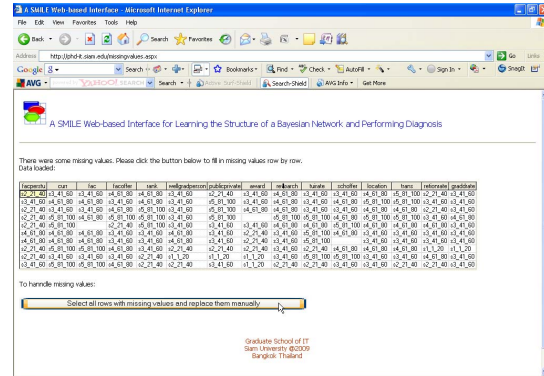


Figure 3. Screenshot of the entire data in datagrid.

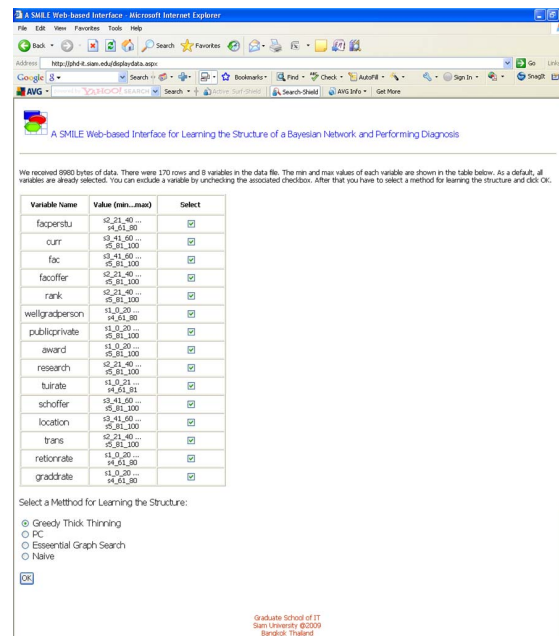


Figure 4. Screenshot of the display of variables and algorithms.

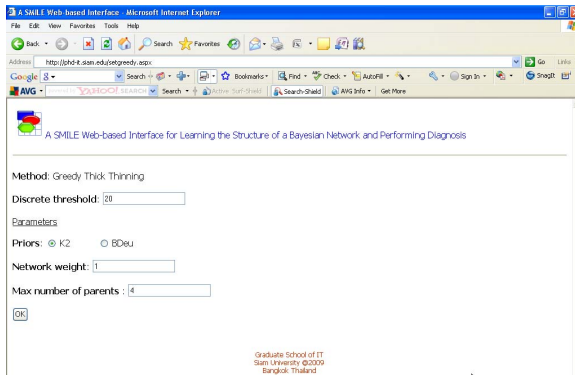


Figure 5. Screenshot of setting parameters for the selected algorithm.

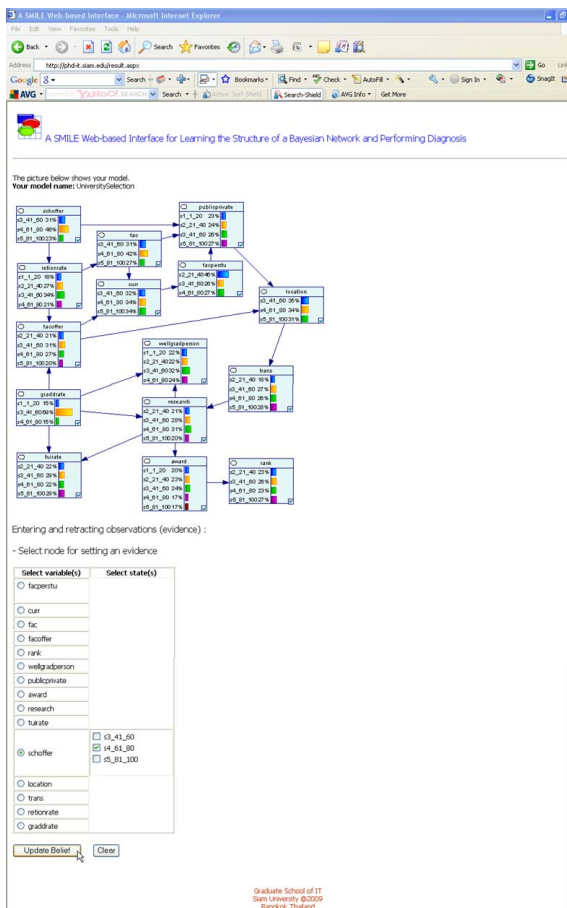


Figure 6. Screenshot of the BN model after updating belief.

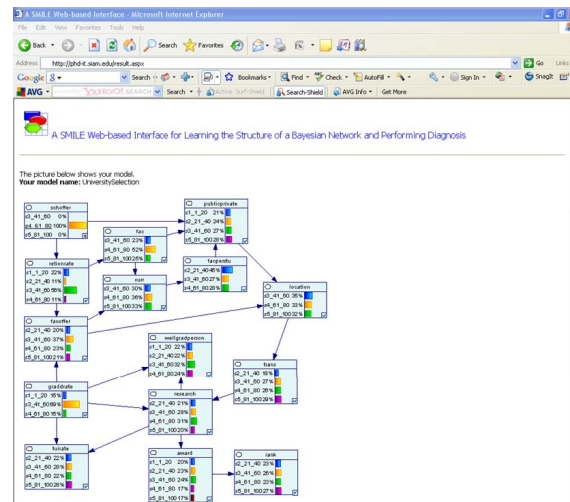


Figure 7. Screenshot of the BN model after setting evidence and updating belief.

## V. CONCLUSION AND FUTURE WORK

This paper presented 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. 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 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 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 in the web page.

The two main principles utilized in the design the SMILE web-based interface for learning the causal structure and performing diagnosis of a Bayesian network were transparency and ease of use. The user just follows a simple six step procedure for constructing his or her BN model. Our future work will focus on improving a decision-oriented diagnosis approach. The SMILE web-based interface has been extended to cope with influence or relevance diagrams. The next version of the SMILE web-based interface will allow users to quantify a decision maker's decision options and preferences and use these to determine an optimal decision policy.

## ACKNOWLEDGMENT

The authors would like to thank the Decision Systems Laboratory, University of Pittsburgh for supporting documents, and source file of the engines: Structural Modeling, Inference, and Learning Engine (SMILE), SMILEarn, and SMILE.NET wrapper. All necessary files and documentations have been obtained from the Decision Systems Laboratory's web site. It is available at <http://genie.sis.pitt.edu>.



## REFERENCES

- [1] R. A. Howard and J. E. Matheson, "Influence diagrams," in *Readings on the Principles and Applications of Decision Analysis*, R. A. Howard and J. E. Matheson, Eds. Menlo Park, CA: Strategic Decisions Group, 1984, pp. 719–762.
- [2] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA: Morgan Kaufmann, 1988.
- [3] F. V. Jensen, *Bayesian Networks and Decision Graphs*. New York: Springer-Verlag, 2001.
- [4] C. Lacave and F. J. Díez, "A review of explanation methods for Bayesian networks," *Knowl. Eng. Rev.*, vol. 17, no. 2, pp. 107–127, Jun. 2002.
- [5] Cheng, J., Greiner, R., Kelly, J., Bell, D., Liu, W., *Learning Bayesian Networks from Data: An Information-Theory Based Approach*, Department of Computing Sciences, University of Alberta, Faculty of Informatics, University of Ulster, 2001.
- [6] Marek J. Druzdzel. 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)*, pages 902–903, Orlando, FL, July 18–22 1999.
- [7] [http://genie.sis.pitt.edu/wiki/SMILE\\_Documentation](http://genie.sis.pitt.edu/wiki/SMILE_Documentation)
- [8] <http://genie.sis.pitt.edu/wiki/SMILearn>
- [9] [http://genie.sis.pitt.edu/wiki/JSMILE\\_and\\_Smile.NET](http://genie.sis.pitt.edu/wiki/JSMILE_and_Smile.NET)
- [10] [http://genie.sis.pitt.edu/wiki/GeNIe\\_Documentation](http://genie.sis.pitt.edu/wiki/GeNIe_Documentation)
- [11] Petri Myllymaki, Tomi Silander, Henry Tirri, and Pekka Uronen, "B-Course a Web-based Tool for Bayesian and Causal Data Analysis," *International Journal on Artificial Intelligence Tools*, vol. 11, No. 3, 2002, pp. 369–387.
- [12] T. Elvira Consortium, "Elvira: An environment for creating and using probabilistic graphical models," in *Proc. 1st Eur. Workshop PGM*, Cuenca, Spain, 2002, pp. 1–11.
- [13] Carmen Lacave, Manuel Luque, and Francisco Javier Díez, "Explanation of Bayesian Networks and Influence Diagrams in Elvira," *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, vol. 37, No. 4, August, 2007.
- [14] Nipat Jongsawat, Pittaya Poompuang, Wichian Premchaiswadi, "Dynamic Data Feed to Bayesian Network Model and SMILE Web Application," 9<sup>th</sup> IEEE International Conference on International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (ACIS 2008), pp. 931–936..
- [15] Friedman, N., The Bayesian Structural EM Algorithm, *Proceedings of 14th Conference Uncertainty in AI*, 1998.
- [16] Saptawati, G.A., Rachmat, H.B., Laksana, D, Construction of Bayesian Network Structure from Data with K2 and B Algorithm, *National Conference on SNIKTI*, 2004.
- [17] Singh, M.; Valtorta, M., Construction of Bayesian, Network Structures from Data: a Brief Survey and an Efficient Algorithm, Dept. of Computer Science, University of South Carolina, Columbia, USA, 1995.
- [18] Spirtes, P., Meek, C., Learning Bayesian Networks with Discrete Variables from Data, *Proceedings of the Conference on Knowledge Discovery & Data Mining*, 1995.
- [19] C.R.T. Thijssen. SmileX: An ActiveX decision-analytics reasoning engine and its application to evaluation of credit applicants. Master's thesis, Delft University of Technology, 1999.
- [20] Pradhan, M.; Provan, G.; Middleton, B.; and Henrion, M. 1994. Knowledge engineering for large belief networks. In *Proceedings of the Tenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-94)*, 484–490.