International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:5, No:7, 2011

# Bayesian Network Model for Students' Laboratory Work Performance Assessment: An Empirical Investigation of the Optimal Construction Approach

Ifeyinwa E. Achumba, Djamel Azzi, and Rinat Khusainov

***Abstract*—**There are three approaches to complete Bayesian Network (BN) model construction: total expert-centred, total data-centred, and semi data-centred. These three approaches constitute the basis of the empirical investigation undertaken and reported in this paper. The objective is to determine, amongst these three approaches, which is the optimal approach for the construction of a BN-based model for the performance assessment of students' laboratory work in a virtual electronic laboratory environment. BN models were constructed using all three approaches, with respect to the focus domain, and compared using a set of optimality criteria. In addition, the impact of the size and source of the training, on the performance of total data-centred and semi data-centred models was investigated. The results of the investigation provide additional insight for BN model constructors and contribute to literature providing supportive evidence for the conceptual feasibility and efficiency of structure and parameter learning from data. In addition, the results highlight other interesting themes.

***Keywords*—**Bayesian networks, model construction, parameter learning, structure learning, performance index, model comparison.

## I. Introduction

A Bayesian Network (BN) model consists of two component parts:

- *network structure* (the qualitative part of the BN) a set of random variables (nodes) and a set of directed edges interconnecting the nodes without creating directed loops, so that the nodes, together with the edges, form a Directed Acyclic Graph (DAG).
- *parameters* (the quantitative part of a BN): value entries in the Conditional Probability Tables (CPTs) associated with each child node in the BN model, which describes the probability distribution of the child node conditioned on every possible combination of the values of its parent nodes, and the Prior Probability Tables (PPTs) associated with each independent node (nodes without parents) in the BN).

Thus, building a BN involves three ordered tasks: identification of the variables and their possible values, identification of the relationships between the variables, and obtainment of the parameters. Each node in a BN represents a random variable or hypothesis and each directed edge represents a relationship (dependency) between the two linked nodes, thereby creating a parent→child relationship.

BN model constructors, ab initio, relied only on domain experts to define both the structure and parameters of a model, currently algorithms exist to construct BN models from data. Hence, BN model construction can be categorized under two approaches: expert- and data-centred. Consequently, there are three techniques to BN model construction:

- *total expert-centred (**tecen**)*
- *total data-centred (**todacen**)*
- *semi data-centred (**sedacen**)*.

In *tecen* approach, the BN model is a product of domain analysis, whereby domain expert(s) completely specify both the qualitative and quantitative components of the model. The *todacen* approach uses algorithms to generate both the qualitative and quantitative components of a BN model from data. The generation of the qualitative component from data is referred to as *structure learning*, and the generation of the quantitative component referred to as *parameter learning*. The *sedacen* approach is a hybrid framework whereby domain expert(s) assist in the creation of the qualitative component of a BN model, while the quantitative component is learnt from data.

A BN-based model, the LAboratory Performance (LAP) model, was constructed with the assistance of domain experts, for the performance assessment of students' laboratory work, in a Virtual Electronic Laboratory (VEL) environment. Detailed descriptions of the VEL and the LAP model, together with their evaluation processes and results are given in [1] and [2] respectively. Having constructed the LAP model with the assistance of three domain experts, there was need to investigate if the model is the best or optimal model for the intended purpose. That is, to empirically investigate if the expert-centred approach is the best approach for constructing the LAP model, or if it is possible to derive an improved or better model using the data-centred BN model construction approach. This required the construction of *sedacen* and *todacen* models from data, with respect to the focus domain. The aim is to compare the performances of the *sedacen* and *todacen* models to the performance of the *tecen* LAP model, based on a set of performance metrics.

There are two possible sources of sample datasets that can be used for the construction of the *sedacen* and *todacen* models. First, sample domain historical dataset(s) on students' laboratory work performance assessment, with respect to a set of performance indicators and their respective criteria, could be used if available. This option was not possible as there are

I. E. Achumba is a research student in the Electronic and Computer Engineering (ECE) Department of University of Portsmouth, UK. (phone: +44 (0)239284 2580; e-mail: Ifeyinwa.Achumba@port.ac.uk)
D. Azzi is a Principal Lecturer in the ECE, Department of University of Portsmouth, UK. (phone: +44 (0)239284 2580; e-mail: Djamel.Azzi@port.ac.uk)

no existing historical sample datasets, for the domain of engineering students' laboratory education (with respect to performance-based assessment of students' laboratory work), to the best of the authors' knowledge. The second option is the use of simulated sample datasets. Often, researchers needing to undertake empirical investigations, with respect to structure and/or parameter learning, create frameworks that would allow them to generate the required sample datasets from a reference model. This approach has been adopted by a number of researchers including [3], [4], [5], [6]. The procedure starts with an existing model (the reference model), generates datasets from the Joint Probability Distribution (JPD) represented by the model, and then uses learning algorithms to attempt to retrieve the reference model from the datasets. The retrieved (learnt) model is then compared with the original model that generated the dataset [6]. This procedure is commonly used for evaluating learning algorithms, and was deemed appropriate for this empirical investigation because of lack of existing historical sample datasets. The idea is that if, using the above procedure, the algorithms fail to retrieve (induce) the reference model, a comparable model, or a better model (in terms of performance) from the sample datasets generated using the JPD encoded by its structure, then it may imply that the algorithms will also fail to induce a comparable model, or a better model from sample datasets generated from other sources. That is, failing to retrieve the reference model, the algorithms should at least learn a model whose performance is comparable to or better than that of the reference model. It is assumed that the reference model is the existing optimal model. If the algorithms learn a model whose performance is significantly better than that of the reference model, then the learnt model is taken to be the optimal model, else the reference model is taken to be the optimal model. Optimal, in this context, refers to the model which is best in terms of the adopted optimality criteria [7].

First, section II details the different BN model construction approaches. Section III describes the procedure for the empirical investigation, highlighting how the different *sedacen* and *todacen* models used in the investigation were constructed, and the model test process, while section IV highlights the criteria and comparative tools used to compare the models. The results of the investigation are presented in section V, and the discussions given in section VI. The paper is summarised in section VII.

## II. BN Model Construction Approaches

### A. Expert-Centred Approach

"Manually" building a BN model involves three ordered tasks: identification of the network variables and their possible values (states), definition of the relationships between the variables, and model calibration (parameterization). There are no formal foundations for "manual" BN model construction, and the process is still essentially an art [8]. It depends on the model constructor to use suitable techniques and tools for undertaking the knowledge elicitation tasks. Expert-centred BN model construction approach offers a number of benefits:

- the model embeds reasonably accurate domain knowledge because it is built interactively with expert(s). The model variables, their states, and relationships are fully appreciated, and the reasoning and rationale behind the BN model can be clearly articulated and communicated.
- model creation is often based on the consensus or average of information and opinions of more than one domain expert, thereby enabling the capture of uncommon or rare scenarios and knowledge.
- the technicalities of the domain represented by the model can be verified/discussed in details at each stage of the development cycle.
- Expert probability elicitation codifies knowledge so that the knowledge is available in the future for other projects and systems thereby promoting reliability in assessment of a family of systems that change within a changing usage environment [9].

However, knowledge elicitation is often said to be a major challenge of the expert-centred BN model construction approach because it is difficult to elicit expert knowledge, which is often biased, and experts rarely agree. Guidelines for easing the elicitation process and elicitation methods have been outlined by [10] [11] [12]. Experts' opinion disagreement is generally acknowledged [13]. Methods for resolving expert opinion conflicts and how to obtain composite or consensus opinion are addressed by [13]. Also, [9] argued that the issue of bias no longer holds as a range of techniques and tools that minimize the effort required for probability elicitation have been developed. In addition, the issue of bias is often addressed through the involvement of more than one domain expert and the knowledge elicitation process often goes through review stages, after which the model is subjected to sensitivity analysis. Moreover, BNs have been found not to be too sensitive to inaccuracies in their parameters [14], so determining good parameter values is in many application areas is quite feasible [15].

### B. Data-Centred Approach

Let $B=\langle G,\theta\rangle$ be a BN model, where $G$ is the network structure with nodes corresponding to the set of random variables, $X=(X_1,\ldots,X_m)$, in the focus domain, and $\theta$ represents the set of parameters for the network. $B$ encodes the JPD $p(X_1,...,X_m)=\prod_{i=1}^{m} p(x_i \mid pa(x_i))$, where $pa(x_i)$ represents the parent set of node $x_i$. The probability distribution, $P(x_i \mid pa(x_i))$, for each discrete node, $X_i$, is represented as a CPT at node $X_i$ in $B$. The data-centred approach entails learning both the structure, $G$, and parameters, $\theta$, from a given sample dataset, or only the parameters.

A dataset, $D$, is a table consisting of records of observations for the network variables, such that, $D=[d_1,d_2,.....,d_N]$, where $N$ = total number of records in $D$, and

$d_l = \{x_1[l], x_2[l], ....., x_m[l]\} \in D$, $l = 1$ to $N$, represents a record of observation for all the variables, $X$. A dataset can be complete or incomplete. A complete dataset contains No Missing Values (NMV) for any of the variables, implying full observability. An incomplete dataset contains missing values for some variables in some or in all the records in the dataset, implying partial observability or presence of latent (hidden) variables, respectively.

*Parameter Learning*

In parameter learning, the structure, $G$, is known and the problem is to learn the parameter, $\theta$, from the given dataset, $D$. That is, the estimation of $\theta = \{\theta_i\}_{i=1,...,m}$, from $D$, given $G$, where $\theta_i$ is the set of numerical value entries in the CPT of node $X_i$. $\theta$ is the complete set of parameters that can best explain the set of observations, $D$ [16]. Parameter learning could involve single or multiple parameters. *Single Parameter Learning* implies that the variable, $X_i$ has only two possible mutually exclusive states denoted, $x_i$ and $\bar{x}_i$, such that the probability mass function $p(X_i)$ is defined by: $p(X_i = x_i) = \theta_i$ and $p(X_i = \bar{x}_i) = 1 - \theta_i$. *Multinomial Parameter Learning* implies that $X_i$ is a multinomial variable with $r > 2$ possible states, $x_{i1}, ....., x_{ir}$, such that $X_i$ has the set of probabilities, $\theta_i = (\theta_{i1}, ....., \theta_{ir})$, respectively, where $\sum_{k=1}^{r} \theta_{ik} = 1$.

*Structure Learning*

Given the dataset, $D$, the structure learning problem is to find, using $D$, the most probable network structure, $G_i$, from among the set of possible network structures, $\Phi = (G_1, G_2, ....., G_\lambda)$, where $\lambda$ is the cardinality of the search space. That is, discover the BN structure that most likely generated $D$. That is, $G_i$ is the network that best describes the conditional independences suggested by the given dataset, $D$ [17]. This is often referred to as *model selection* in literature, which term will not be used in this paper because, in this context, BN model refers to a complete Bayesian network (structure + parameters). Once $G$ is found, its parameters, $\theta$, are derived as described earlier.

Structure learning algorithms are either based on Conditional Independence (CI) tests or Search and Score (SaS). The CI approach uses *constraint-based* algorithms to find the structure whose implied independence constraints "match" those found in the data by performing CI tests on tuples of variables, using *statistical tests* or *information theoretic* measures [18]. CI-based algorithms include the PC algorithm by [19]. The SaS approach consists of three components: the search space, the scoring function, and the search engine. The *search space* consists of the set of all possible BN structures, $\Phi$, given the domain variables. The main operation in the search space is the modification of one structure to produce another structure with the operators ''add an edge'', ''delete an edge'', and "reverse an edge" [20].

The *score metric* takes the dataset and a possible structure and returns a score reflecting the goodness-of-fit of the data to the structure [21]. There are two categories of scoring functions: Bayesian and information-theoretic. The information-theoretic score functions include: the Log-likelihood (LL) [22], Minimal Description Length (MDL) [23], Akaike Information Criterion (AIC) [24], and the Bayesian Information Criterion (BIC) [25]. The MDL is said to be equivalent to the BIC function; hence they are often written as MDL/BIC. The Bayesian scoring metrics include: Bayesian Dirichlet (BD) [26], likelihood-equivalence Bayesian Dirichlet (BDe) [26], the uniform joint distribution Bayesian Dirichlet (BDeu) [27], and the K2 [3]. The K2 has been described as one of the most successful scoring metrics [28].

The *search engine* (*search algorithm*) works to identify structures with high scores by exploring the search space. It makes comparisons of network structures as it searches heuristically for the most likely structure [29]. Essentially, the dataset $D$, the scoring function, and the search space constitute the inputs to the search algorithm while the output is a network that maximizes the score, $P(D \mid G_i)$, the probability of the most probable structure, $G_i$, given the dataset, $D$ [30].

One of the main challenges of the data-centred approach is that structure learning is NP-hard [31]. Researchers have attempted to reduce the complexity of BN structure learning by various algorithmic means, but the problem remains complex and hard, without exact and exhaustive solution [18]. Consequently, heuristic algorithms are often employed for the learning process. The latter help produce an acceptable solution to a problem in many practical scenarios, though it is not certain to arrive at an optimal solution. It begins with an approximate method of solving the problem within the context of the goal, and then uses feedback from the solution to improve its performance, searching for a satisfactory solution rather than optimal solution.

The complexity of the search space is another major challenge of structure learning because the number of possible structures grows super-exponentially with the number of variables, $n$, in the problem domain [18]. For, $n$ variables, the cardinality of the search space is given by [32] as the recursive function:

$$f(n) = \sum_{k=1}^{n} (-1)^{k+1} \binom{n}{k} 2^{k(n-k)} f(n-k) \quad , \quad \text{where}$$

$f(1) = 1$

Table 1 lists the possible number of BN structures for some values of $n$.

TABLE I
NUMBER OF POSSIBLE BN STRUCTURES FOR VARIOUS NUMBERS OF VARIABLES

| No of variables, *n* | No of possible BN structures |
|---|---|
| 1 | 1 |
| 2 | 3 |
| 3 | 25 |
| 4 | 543 |
| 5 | 29,281 |
| 6 | 3,781,503 |
| 7 | 1,138,779,265 |
| 8 | 78,370,2329,343 |
| 9 | 1,213,442,454,842,881 |
| 10 | 4,175,098,976,430,598,100 |

This super-exponential relationship between the number of variables and the number of possible structures is a major source of computational complexity [33].

## III. INVESTIGATION PROCEEDURE

First, sample datasets were generated with the reference model in such a way that the investigation was undertaken from two different approaches referred to, in this context, as *non-parameteric and parameteric* approaches. The *non-parameteric* approach entailed the use of the unparameterised version of the reference model (structure without the elicited parameters) to generate some of the sample datasets for the investigation. The *parameteric* approach entailed the use of the parameterised version of the reference model (structure with elicited parameters) to generate some of the sample datasets for the investigation. A set of four training (learning) datasets were generated using each of the two approaches. Hence, there were a total of eight training datasets, $Dnp = \left(D1np, D2np, D3np, D4np\right)$ and $Dp = \left(D1p, D2p, D3p, D4p\right)$. The *np* and *p* at the end of the dataset names indicate *non-parameteric* and *parameteric* approaches to the generation of the datasets, respectively. Models constructed with training datasets generated using the non-parameteric approach are referred, in this context, as *non-parameteric models*, while models constructed with training datasets generated using the *parameteric* approach are referred as *parameteric models*. Also, a set of four test datasets, $T = (T1, T2, T3, T4)$, different from the training datasets, were generated for the evaluation of the models. Several training datasets were used to facilitate the acquisition of meaningful data for the intended analysis.

The BN software tools, Genie [34], Netica [35], SamIam [36], and WinMine Toolkit [37] were used for the investigation. Genie is the graphical interface to SMILE [34], a Bayesian inference engine. Netica is a complete program for working with belief networks. SamIam (Sensitivity Analysis Modeling Inference And More) is a tool for modelling and reasoning with Bayesian networks, developed in Java, and includes a GUI for editing Bayesian networks.

### A. Construction of the Data-centred Models used for the Investigation

A total of twenty four (24) *sedacen* models were constructed, consisting of three sets of four *non-parameteric* models, $PNDnp = \left(PND1np, PND2np, DPN3np, PND4np\right)$, $PGDnp = \left(PGD1np, PGD2np, PGD3np, PGD4np\right)$, and $PSDnp = \left(PSD1np, PSD2np, PSD3np, PSD4np\right)$; and three sets of four *parameteric* models, $PNDp = \left(PND1p, PND2p, DPN3p, PND4p\right)$, $PGDp = \left(PGD1p, PGD2p, PGD3p, PGD4p\right)$, and $PSDp = \left(PSD1p, PSD2p, PSD3p, PSD4p\right)$.

The model names indicate the type of learning undertaken, and the software tool and the dataset used for the construction of the model. For example, $PND1np$ indicates that the model was constructed by parameterizing, using Netica, a known structure from the training dataset, $D1np$. Netica, Genie and SamIam were used for the construction of the *sedacen* models, based on the Expectation Maximization (EM) algorithm, implemented by all three software tools. The plan to also undertake parameter learning using Gibbs sampler, for comparative purposes, was jettisoned because, according to [38], EM and Gibbs sampler are substantially equivalent in their parameter estimates.

It was possible to use the same sets of datasets with the three different software tools, Netica, Genie, and SamIam, because they all support text data file formats, albeit with different file extensions (.cas, .txt/.dat, and .dat, respectively). All that was required was to save the data file with the appropriate file extension for each software tool and edit as may be necessary (for example, Netica requires the inclusion of the row number for each record of the data file, while Genie and SamIam do not) . Also, all the software tools used similar network file formats (.dne, .dnet, and .net) so it was possible to open any of the networks with any of the tools.

It was aimed to construct a total of thirty two (32) *todacen* models from the two sets of four training datasets, $Dnp = \left(D1np, D2np, D3np, D4np\right)$ and $Dp = \left(D1p, D2p, D3p, D4p\right)$, using Genie and WinMine. WinMine supports SaS-based structure learning approach, while Genie supports both CI- and SaS-based. Exploiting this, in order to exact more investigative effort at inducing a better model than the reference model, the *todacen* model construction using Genie was done as per the following three steps:

a. first, eight *todacen* models were constructed based on the CI approach with the PC algorithm, using each training datasets in the two sets of training datasets. Search time limit was not imposed. This step generated the $SPGaDnp = \left(SPGaD1np, SPGaD2np, SPGaD3np, SPGaD4np\right)$ and $SPGaDp = \left(SPGaD1p, SPGaD2p, SPGaD3p, SPGaD4p\right)$ groups of models. The model name, $SPGaD1np$, for example, indicates that the model was constructed by

learning both structure and parameters, using Genie, this step, a, and the training dataset, $D1np$ .

b. next, another eight models were constructed with the Greedy Thick Thinning (GTT) search algorithm and the K2 score metric, based on the SaS approach, using each training datasets in the two sets of training datasets to get the
$SPGbDnp = (SPGbD1np, SPGbD2np, SPGbD3np, SPGbD4np)$
and $SPGbDp = (SPGbD1p, SPGbD2p, SPGbD3p, SPGbD4p)$
groups of models. The model name, $SPGbD1np$ , for example, indicates that the model was constructed by learning both structure and parameters, using Genie, this step, b, and the training dataset, $D1np$ .

c. finally, eight models were constructed with the Greedy Thick Thinning (GTT) search algorithm and the BDeu score metric, based on the SaS approach using each training datasets in the two sets of training datasets to get
$SPGcDp = (SPGcD1p, SPGcD2p, SPGcD3p, SPGcD4p)$
groups of models. The model name, $SPGcD1np$ , for example, indicates that the model was constructed by learning both structure and parameters, using Genie, this step, c, and the training dataset, $D1np$ .

Factors related to the learning algorithms and score metrics, such as maximum adjacency size (for the PC algorithm) and number of parents constraints (for the K2 and BDeu score metrics), were varied, aimed at increasing the chances of inducing a model structure that will yield a better model than the reference model.

Furthermore, two groups of models, $SPWDnp = (SPWND1np, SPWD2np, SPWD3np, SPWD4np)$ and $SPWDp = (SPWD1p, SPWD2p, SPWD3p, SPWD4p)$, were constructed from the two sets of four training datasets, using WinMine Toolkit. The letters, *SP,* in the model names imply structure and parameter learning, *W* implies WinMine Toolkit, and $D1np$ and $D1p$ highlight the particular training dataset used for the construction of the model.

### B. Evaluation of the Models

The reference, *sedacen,* and *todacen* models were evaluated using the set of four test datasets. The test procedure consisted of entering findings at selected evidence nodes of a model, and querying one or more target nodes. The nodes of the model are divided into two sets: evidence and target nodes. Any node can belong to any one of the two sets, for the purposes of the test. It is often preferable to choose as target, the node that in the real context would be target of inference. The values in each record of the test dataset are split into two sets: values for the chosen evidence nodes, and values for the chosen target nodes. The values for the evidence nodes are entered as findings into the network, the network is updated, and inference made at the target nodes. This process is repeated for each record in the test dataset. For each network update, the probability distribution of the target node is recorded and its prediction determined. That is, after each network update, the state with higher belief value (the most likely or maximum likelihood state), based on a cut-off threshold probability, is taken to be the prediction for the target node. For example, for a 50% cut-off threshold probability, of the two states of the target node, the state which belief level is higher than 50% is taken to be the prediction. The predictions are then compared with the observations (the set of values for the target node in the test dataset taken as the actual "observations"), for each record of the test dataset. If a prediction corresponds to the observation, it is recorded as a success otherwise it is recorded as a failure. The statistics are then collected and used to assess the performance of the model, generating values for the various performance metrics that constitute the optimality criteria.

## IV. Optimality Criteria and Comparative Tools

### A. Structure Comparison

Structural difference measures are often used to compare the structural differences between an induced model structure and a reference model structure [6]. The comparison may sometimes not take into consideration the orientation of the edges. If the focus is causality, then the orientation of the edges becomes extremely important. Otherwise, the orientation of some of the edges can be deemphasized [6]. A causal model is a "Bayesian network with added property that the parents of each node are its direct causes" [39]. This implies an asymmetric relationship between parent and child nodes, such that in the case edge of reversal, the resulting network will not be equivalent in terms of representational ability. In induced non-causal model structures, it is possible to ignore the direction of the reversible edges but not those of the compelled edges. The reversible edges are the edges that occur in the opposite direction in some other DAG that is equivalent (in terms of representational ability) to the current DAG [40]. "If two DAGs encode the same conditional independencies, they are called Markov equivalent. The set of all DAGs can be partitioned into Markov equivalence classes. Graphs within the same class can have the direction of some of their arcs reversed without changing any of the CI relationships. Each class can be represented by a PDAG (partially directed acyclic graph) called an essential graph or pattern. This specifies which edges must be oriented in a certain direction (compelled edges), and which are reversible. When learning graph structure from observational data, the best one can hope to do is to identify the model up to Markov equivalence" [41].

In this context, ignoring the reversible edges, the link statistics of an induced model structure are categorized as:

- *correct positive* (*cp*)-- a link is learnt between two nodes where a link exists between the same two nodes in the reference model (correct link)
- *false positive* (*fp*)-- a link is learnt between two nodes where a link does not exist between the same two nodes in the reference model (extra link)
- *correct negative* (*cn*)-- no link is learnt between two nodes where a link does not exist between the same two nodes in the reference model (correct nolink)

- *false negative* (*fn*)-- no link was learnt between two nodes where a link exists between the same two nodes in the reference model (missing link).

### B. Performance Metrics

Scoring functions or rules are appropriate for evaluating the performance of probabilistic predictive models [42]. Mathematically convenient scoring rules are most commonly used [43]. These include, *error rate, logarithmic* (*logloss*) *score,* and *Briers score* [44] [42] [45][18]**.** *Sensitivity* is also, often used as a model performance and comparison measure [18][45].

The error rate, based on the maximum likelihood state of the target node [46], is a way to analyze model predictions by dividing the number of predictive errors by the number of test cases in the test dataset. It gives the percentage failure rate. It identifies the percentage of the cases in a test dataset for which the network predicted a wrong value for the query node. For example, an error rate of 24% implies that in 24% of the cases for which the test dataset contains a value for the target node, the predictions did not match the observed values.

The logarithmic (logloss) score was suggested by [47] and is defined as follows: let *X* denote a discrete random variable, with *m* (mutually exclusive) possible states, $(x_1, x_2, \ldots, x_i \ldots, x_m)$, which is to be observed for a sequence of cases, $i$ = 1,……..,*N*. Let $p(x_i)$ denote the estimated probability (referred to as the predicted value for the purposes of the test) for the $i^{th}$ state. Suppose the $j^{th}$ state is actually observed, then the particular observation is associated with a logloss score for the $j^{th}$ state given by [Cowell, 1999b] [Jenson 2001] as: $\ell_j = \log \frac{1}{p(x_j)} = -\log p(x_j)$. Then, by accumulating the scores for the *N* cases, a total penalty for the *N* observations is obtained by: $\ell = \sum_{j=1}^{N} \ell_j$, and the average logloss score for the *N* cases is: $\ell_{avg} = \frac{1}{N}\sum_{j=1}^{N} \ell_j = \frac{1}{N}\sum_{j=1}^{N} -\log p(x_j)$. The logloss value lies in the range [0, ∞], where smaller (lower) values of the score imply better model performance.

The Brier score (*b*), also referred to as Quadratic Loss (QL) or Mean Squared Error of Prediction (MSEP), measures the accuracy of a set of probability assessments. The Brier score function, as used in BN model performance comparison, is given by [17][18] [45][48] as:

$$b = \frac{1}{N}\left[\sum_{i=1}^{N}\left(\left(1 - 2 \times p(y = c \mid x_i)\right) + \sum_{j=1}^{k} p(y = j \mid x_i)^2\right)\right]$$

where $p(y = c \mid x_i)$ is the probability predicted for the actual (observed) state, $c$, of the target variable, *y* (the state of *y* in the particular record of the test dataset), given the evidence variables, $x_i$; $p(y = j \mid x_i)$ is the probability predicted for the $j^{th}$ state of y, given the evidence variables; *k* is the number of states of the target variable, *y*; *N* is the number of records in the test dataset. The QL is a measure of the average quadratic loss that occurred on each instance in the test dataset. It is averaged over all the records in the test dataset and not only accounts for the probability assigned to the actual (observed) state, but also the probabilities assigned to the other possible states of *y*. The value of Brier score lies in the range [0, 1], with $b$ = 0 indicating higher prediction accuracy, thus better performance.

Sensitivity (also referred to as the recall rate) is a statistical measure of model performance. It measures the proportion (in percentage) of actual values (observations) which are correctly predicted. A sensitivity of 100% means that the model correctly predicted all actual observations for the target variables (100% actual or true positives).

These metrics are often used together, in any one investigation, by researchers [18][45], in order to facilitate the drawing of more robust conclusions. The different metrics, though not complementary, evaluate performance from different perspectives, thereby collectively giving a more robust picture of the performance of a model. Error rate informs on the percentage failure rate of a model, the Brier score gives a measure of the accuracy of the probability estimates made by the model, and sensitivity informs on the percentage success rate of the model. The logloss score is similar to Brier score, however, the logloss score is *local* in that it only depends upon the probability assigned to the particular state and not on any of the probabilities assigned to the other states [49].

## V. Outcome of the Investigation

As stated earlier (in section III), the models constructed for the purposes of this investigation are broadly grouped as *parameteric* and *non-parameteric* models (based on the training dataset used for the construction model), and the type of model (*sedacen* and *todacen*). Table 2 highlights the different sample training and test datasets and their respective sizes.

TABLE II
TRAINING AND TEST DATASETS

| Type of Dataset | Name of set of Datasets | Member Datasets | Sizes of the Datasets |
|---|---|---|---|
| Training | Dp | D1p, D2p, D3p, D4p | 24000, 72000, 142000, 240000, respectively |
| | Dnp | D1np, D2np, D3np, D4np | 24000, 72000, 142000, 240000, respectively |
| Test | T | T1, T2, T3, T4 | 270, 5000, 35000, 70000, respectively |

The sizes of the training datasets were chosen to represent an increasing reasonably spaced size range, for the purposes of the investigation. Full observability was assumed, for the purposes of the investigation. It was also assumed that the data samples, generated using a BN software tool, are representative of the larger set of baselines samples. As highlighted in Table 2, four different sizes of test datasets were used for evaluating the models. One of the reasons was to investigate the relationship between model performance and the size of the test dataset. The second reason was for

repeatability of the test for evaluating the performances of the models in order to facilitate the drawing of more robust conclusions from the investigation. The results of the empirical investigation are hereby presented with respect to the models' performance indices. Rather than use the performance metric values (error rate, logloss, Brier score, and sensitivity) individually, for each of the four test instances, to compare the models, a single performance index was derived for each model, based on the metrics. The function, $\psi$, for calculating the performance index of a model is defined, in this context, is defined as: $\psi = \left[(100-e)+(1-b)*100+s+(1-l)*100\right]_{normalized}$, where $e$ = error rate, $b$ = brier score, $l$ = logloss, and $s$ = sensitivity. The function, $\psi$, takes the values of the performance metrics for a model as input, and yields a performance index for the model, for a test instance. The function assumes equal importance for all the performance metrics. Table 3 lists the performance indices for the *parameteric* models, with respect to the four test instances, while Figure 1 graphically highlights the average performance indices of the *parameteric* models and that of the Reference model. Also, Table 4 lists the performance indices for the *non-parameteric* models, with respect to the four test instances, while Figure 2 graphically highlights the average performance indices of the *non-parameteric* models and that of the Reference model.Tables III and IV, and Figures 1 and 2, highlight performance differences between the parameteric and non-parameteric models, relative to the performance of the Reference model. The performance indices of the SPWDp group of models were not listed in Table 3 because they could not be evaluated with respect to the performance metrics, using WinMine toolkit. The WinMine BN network file format (.xmod) did not allow for its conversion to network file formats supported by other software tools that facilitate model evaluation with respect to the performance metrics. Also, the performance indices of the *non-parameteric* todacen groups of models (with the exception of one) were not listed in Table 4 because no structures or meaningful structures were learnt, hence no model to evaluate.

TABLE III
*PARAMETERIC* MODELS: PERFORMANCE INDICES

| ***PARAMETERIC* MODELS** | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Type** | **Model Group and (Software Tool)** | **Model** | **Structure Learning Approach** | **No of links learnt** | **Performance Index** | | | | |
| | | | | | **T1** | **T2** | **T3** | **T4** | **Average Performance Index** |
| *tecen* | - | REF | - | - | 0.638 | 0.623 | 0.633 | 0.629 | 0.630 |
| *Sedacen* (known structure, parameter learning) | PNDp (Netica) | PND1p | - | - | 0.633 | 0.629 | 0.628 | 0.630 | 0.630 |
| | | PND2p | - | - | 0.633 | 0.629 | 0.629 | 0.630 | 0.630 |
| | | PND3p | - | - | 0.633 | 0.629 | 0.629 | 0.630 | 0.630 |
| | | PND4p | - | - | 0.638 | 0.623 | 0.633 | 0.629 | 0.631 |
| | PGDp (Genie) | PGD1p | - | - | 0.633 | 0.629 | 0.629 | 0.630 | 0.630 |
| | | PGD2p | - | - | 0.638 | 0.623 | 0.633 | 0.629 | 0.630 |
| | | PGD3p | - | - | 0.638 | 0.623 | 0.633 | 0.629 | 0.630 |
| | | PGD4p | - | - | 0.638 | 0.622 | 0.633 | 0.629 | 0.630 |
| | PSDp (SamIam) | PSD1p | - | - | 0.461 | 0.479 | 0.489 | 0.490 | 0.480 |
| | | PSD2p | - | - | 0.328 | 0.326 | 0.327 | 0.326 | 0.326 |
| | | PSD3p | - | - | 0.328 | 0.326 | 0.327 | 0.326 | 0.326 |
| | | PSD4p | - | - | 0.328 | 0.326 | 0.327 | 0.326 | 0.326 |
| *todacen* (structure and parameter learning) | SPGaDp (Genie) | SPGaD1p | CI-Test (PC) | 35 | 0.638 | 0.622 | 0.633 | 0.629 | 0.630 |
| | | SPGaD2p | CI-Test (PC) | 35 | 0.643 | 0.623 | 0.633 | 0.629 | 0.632 |
| | | SPGaD3p | CI-Test (PC) | 35 | 0.638 | 0.623 | 0.633 | 0.629 | 0.630 |
| | | SPGaD4p | CI-Test (PC) | 35 | 0.637 | 0.622 | 0.632 | 0.628 | 0.630 |
| | SPGbDp (Genie) | SPGbD1p | SaS (GTT/K2) | 35 | 0.637 | 0.622 | 0.632 | 0.628 | 0.630 |
| | | SPGbD2p | SaS (GTT/K2) | 35 | 0.637 | 0.623 | 0.633 | 0.629 | 0.630 |
| | | SPGbD3p | SaS (GTT/K2) | 35 | 0.638 | 0.623 | 0.633 | 0.629 | 0.630 |
| | | SPGbD4p | SaS (GTT/K2) | 35 | 0.637 | 0.622 | 0.632 | 0.628 | 0.630 |
| | SPGcDp (Genie) | SPGcD1p | SaS (GTT/BDu) | 35 | 0.637 | 0.622 | 0.632 | 0.628 | 0.630 |
| | | SPGcD2p | SaS (GTT/BDu) | 35 | 0.637 | 0.621 | 0.632 | 0.628 | 0.629 |
| | | SPGcD3p | SaS (GTT/BDu) | 35 | 0.636 | 0.622 | 0.632 | 0.628 | 0.630 |
| | | SPGcD4p | SaS (GTT/BDu) | 35 | 0.638 | 0.623 | 0.633 | 0.629 | 0.631 |
| | SPWDp (WinMine Toolkit) | SPWD1p | SaS (?/BDu) | 35 | * | * | * | * | * |
| | | SPWD1p | SaS (?/BDu) | 35 | * | * | * | * | * |
| | | SPWD1p | SaS (?/BDu) | 35 | * | * | * | * | * |
| | | SPWD1p | SaS (?/BDu) | 35 | * | * | * | * | * |

* Could not evaluate the models based on the performance metrics, using the WinMine toolkit.

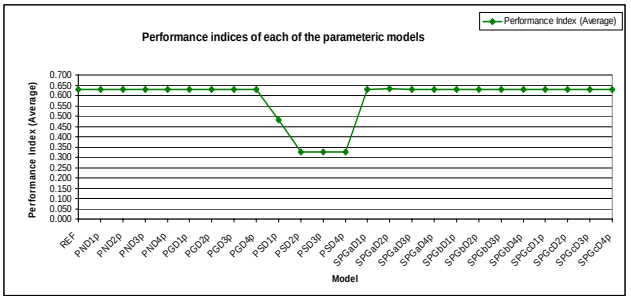


Fig. 1 Average performance indices of the *parameteric* models and the Reference model

TABLE IV
NON-PARAMETERIC MODELS: PERFORMANCE INDICES

| *NON-PARAMETERIC* MODELS | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Type | Model Group and (Software Tool) | Model | Structure Learning Approach | No of links learnt | Performance Index | | | | |
| | | | | | T1 | T2 | T3 | T4 | Average Performance Index |
| *tecen* | - | REF | - | - | 0.638 | 0.623 | 0.633 | 0.629 | 0.630 |
| *Sedacen* (known structure, parameter learning) | PNDnp (Netica) | PND1np | - | - | 0.480 | 0.469 | 0.469 | 0.471 | 0.472 |
| | | PND2np | - | - | 0.436 | 0.438 | 0.433 | 0.431 | 0.435 |
| | | PND3np | - | - | 0.484 | 0.487 | 0.488 | 0.490 | 0.487 |
| | | PND4np | - | - | 0.436 | 0.456 | 0.453 | 0.454 | 0.450 |
| | PGDnp (Genie) | PGD1np | - | - | 0.480 | 0.469 | 0.469 | 0.469 | 0.472 |
| | | PGD2np | - | - | 0.436 | 0.455 | 0.453 | 0.454 | 0.449 |
| | | PGD3np | - | - | 0.436 | 0.438 | 0.433 | 0.431 | 0.435 |
| | | PGD4np | - | - | 0.484 | 0.487 | 0.488 | 0.490 | 0.487 |
| | PSDnp (Samlam) | PSD1np | - | - | 0.328 | 0.326 | 0.327 | 0.326 | 0.326 |
| | | PSD2np | - | - | 0.328 | 0.326 | 0.327 | 0.326 | 0.326 |
| | | PSD3np | - | - | 0.328 | 0.326 | 0.327 | 0.326 | 0.326 |
| | | PSD4np | - | - | 0.328 | 0.326 | 0.327 | 0.326 | 0.326 |
| *todacen* (structure and parameter learning) | SPGaDnp (Genie) | SPGaD1np | CI-Test (PC) | 6 (4 nets) | * | * | * | * | * |
| | | SPGaD2np | CI-Test (PC) | 9 (7 nets) | * | * | * | * | * |
| | | SPGaD3np | CI-Test (PC) | 10 (8 nets) | * | * | * | * | * |
| | | SPGaD4np | CI-Test (PC) | 25 (3 nets) | 0.495 | 0.488 | 0.493 | 0.493 | 0.493 |
| | SPGbDnp (Genie) | SPGbD1np | SaS (GTT/K2) | 0 | * | * | * | * | * |
| | | SPGbD2np | SaS (GTT/K2) | 0 | * | * | * | * | * |
| | | SPGbD3np | SaS (GTT/K2) | 0 | * | * | * | * | * |
| | | SPGbD4np | SaS (GTT/K2) | 0 | * | * | * | * | * |
| | SPGcDnp (Genie) | SPGcD1np | SaS (GTT/BDu) | 0 | * | * | * | * | * |
| | | SPGcD2np | SaS (GTT/BDu) | 0 | * | * | * | * | * |
| | | SPGcD3np | SaS (GTT/BDu) | 0 | * | * | * | * | * |
| | | SPGcD4np | SaS (GTT/BDu) | 0 | * | * | * | * | * |
| | SPWDnp (WinMine Toolkit) | SPWD1np | SaS (?/BDu) | 0 | * | * | * | * | * |
| | | SPWD2np | SaS (?/BDu) | 0 | * | * | * | * | * |
| | | SPWD3np | SaS (?/BDu) | 0 | * | * | * | * | * |
| | | SPWD4np | SaS (?/BDu) | 0 | * | * | * | * | * |

* No structure or meaningful structure was learnt, hence no model to evaluate

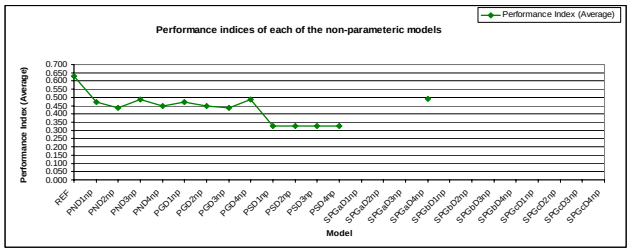


Fig. 2 Average performance indices of the *non-parameteric* models and the Reference model

It was observed, as highlighted and indicated by Tables 5.3 and 5.4, and Figures 5.2 and 5.3, that the:

- best performing model is the SPGaD2p with an average performance index of 0.632, as against the Reference model's average performance index of 0.630.
- performances of the *parameteric* groups of model (with the exception of one) are comparable (equivalent) to the performance of the Reference model. That is, the performances of 67% of the *parameteric sedacen* and 100% of the *parameteric todacen* models were comparable (equivalent) to that of the reference model.
- The performances of the *non-parameteric* models were relatively poor compared to the performance of the Reference model. That is, taking 0.500 as the threshold between comparable and poor performance, the performances of 100% of both the *non-parameteric sedacen* and *todacen* models were relatively poor. The learnt CPT entries were more or less inconclusive.
- In 15 (94%) of the 16 cases in which the *Dnp* set of datasets (*non-parameteric* datasets) were used for the construction of *todacen* models (structure and parameter learning), no structures or meaningful structures were learnt.

In addition, a weak negative correlation ($r$ = -0.0734) was found between the size of the training dataset and model performance. Also, a weak negative correlation ($r$ = -0.0569, $p$ = 0.94400) was found between the size of the test dataset and model performance.

## VI. DISCUSSION

The results of the empirical investigation are encouraging and contribute to the literature providing supportive evidence for the conceptual feasibility and efficiency of structure and parameter learning algorithms and approaches. The induction of models which performances were comparable (equivalent) to, and in one case better than (albeit marginally by 0.002 (0.32%)) the performance of the Reference model is significant. However, the results show that it may not have been possible to construct the *sedacen* and *todacen* models with performances that are comparable to or better than the performance of the Reference model without first constructing the complete Reference model (structure + parameters), with assistance of domain experts. Though the performances of all the models (*parameteric* and *non-parameteric, sedacen* and *todacen*) constructed using the SamIam software tool were relatively poor, it is assumed to imply inefficiency of the software tool, which however, may be due to some uncontrolled factor(s) and therefore may require further investigation. The results also indicate that the Conditional Independence (CI) test based PC structure algorithm is equivalent in its learning outcomes to the Score and Search (SaS) based GTT/K2 and GTT/BDeu structure learning algorithms, with respect to the *parameteric* models. The PC algorithm performed better than the SaS-based algorithms with respect to the *non-parameteric* models. It was able to learn some links (in 100% of the cases), though the links did not yield meaningful models in 75% of the cases. Furthermore, the sizes of the training and test datasets did not seem to have any relationship with model performance. However, the results showed that in 100% of the 12 *non-parameteric sedacen* models, almost equal probability values were assigned to all possible parent combinations. The parameter values seemed logically and realistically unacceptable for the purpose for which the models are aimed, as highlighted in Figure 5.4, the CPT of one of the nodes in a *non-parameteric sedacen* model.

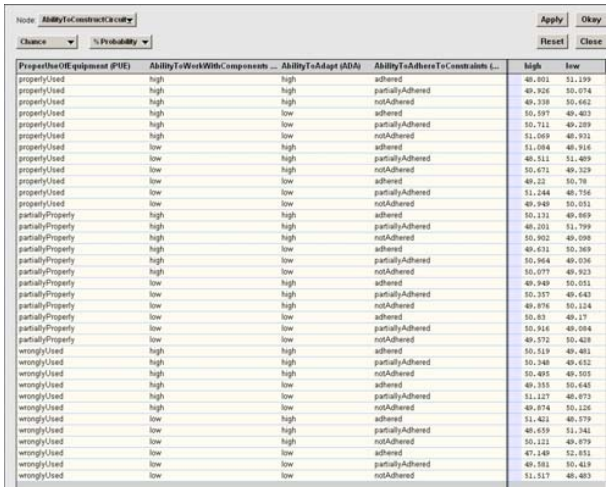


Fig. 3 The CPT of a node in a non-parameteric sedacen model, PND2np

Finally, the results have also highlighted an important area that may require further investigation. The results showed that a complete Reference model (that is knowledge of the relationship between the domain variables and their Conditional Probability Distributions) is a requirement for simulating sample datasets for structure and/or parameter that will yield meaningful and comparable models for the domain. This suggests the need for further research in order to investigate the outcome of structure and/or parameter learning using historical sample datasets from the domain that may not have been generated with knowledge of the relationship between the domain variables and their Conditional Probability Distributions. The main challenge to this investigation will be the obtainment of historical sample datasets for the domain.

## VII. Conclusion

The best approach for the construction of the BN-based model for the performance assessment of students' laboratory work in the VEL environment has been empirically investigated. The optimisation exercise has yielded a, albeit marginally, better model. This provides reassurance that the procedure followed in the derivation of the assessment model was fit for purpose. The results additional insight for BN model constructors and contribute to the literature providing supportive evidence for the conceptual feasibility and efficiency of structure and parameter learning algorithms and approaches. In addition, they also highlighted the need for further investigation with respect to data-centred BN model construction approach for the domain. Furthermore, from our experience, the data-centred BN model construction approach depends on the availability of appropriate software tools and sample training datasets. Model construction may be limited by the software tools available. Commercial software tools may be inaccessible, in which case freeware tools, which may have limited capabilities, are used. Also, there seems to be no standardized data and network file formats for BN software tools. Different tools support different data and network file formats. For example, some software tools may support only numeric data files, some string data files, while some may require the inclusion of record occurrence frequencies. This may be counterproductive for the data-centred BN construction approach.

## References

[1] D. Azzi, I. E. Achumba, V. L. Dunn, and G. A. Chukwudebe, "Intelligent Performance Assessment of Students' Laboratory Work in a Virtual Electronic Laboratory Environment", Paper submitted to *IEEE Transactions on Learning Technologies*, on 09/Dec./2010, for review.

[2] I. E. Achumba, and D. Azzi, "A Virtual Electronic Laboratory for Genuine Practical Experiences: Description and Evaluation", Paper submitted to *IEEE Transactions on Learning Technologies*, for review. Revised and resubmitted on 09/Dec./2010.

[3] G. F. Cooper and E. Herskovits, "A Bayesian Method for the Induction of Probabilistic Networks from Data", *Machine Learning*, vol. 9, no. 4, pp. 309–347.

[4] N. Friedman, "The Bayesian Structural EM Algorithm", *Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence (UAI-98)*, 1998, pp. 129 – 138.

[5] Y. Pan and E. S. Burnside, "The Effects of Training Parameters on Learning a Probabilistic Expert System for Mammography", *International Congress Series*, vol. 1268, 2004, pp. 1027 – 1032.

[6] M. de Jongh, M. and M. J. Druzdzel, "A Comparison of Structural Distance Measures for Causal Bayesian Network Models", *Recent Advances in Intelligent Information Systems*, 2009, pp. 443–456. Retrieved September 07, 2010, from http://iis.ipipan.waw.pl/2009/proceedings/iis09-43.pdf.

[7] L. A. ZADEH, "What is Optimal?", *IRE Transactions on Information Theory,* 1958, pp. 3. Retrieved September 08, 2010, from http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=01057441.

[8] M. J. Druzdzel and H. A. Simon, "Causality in Bayesian Belief Networks", *Proceedings of the Conference on Uncertainty in Artificial Intelligence,* 2006, pp. 3-11.

[9] N. Fenton and M. Neil, "Comment: Expert Elicitation for Reliable System Design", *Statistical Science*, vol. 21, no. 4, 2006, pp. 451 – 453.

[10] R. Rajabally, P. Sen, S. Whittle, and J. Dalton, "Aids to Bayesian Belief Network Construction", *Proceedings of the 2nd International IEEE Conference on Intelligent Systems*, vol. 2, 2004, pp. 457 – 461.

[11] B. Das, "Generating Conditional Probabilities for Bayesian Networks: Easing the Knowledge Acquisition Problem", August 4, 2008. Retrieved on February 24, 2010, from http://arxiv.org/ftp/cs/papers/0411/0411034.pdf.

[12] Druzdzel, M. J., & van der Gaag, L. C. (2000). Building Probabilistic Networks: Where Do the Numbers Come from? (Guest Editors' Introduction). *IEEE Transactions on Knowledge and Data Engineering,* 12(4), 1-485.

[13] K. Ng and B. Abramson, "Consensus Diagnosis: A Simulation Study", *IEEE Trans on Systems, Man, and Cybernetics,* vol. 22, no. 5, 1992, pp. 916 – 928.

[14] M. J. Druzdzel and A. Onisko, "Are Bayesian Networks Sensitive to Precision of Their Parameters?", *Intelligent Information Systems*, 2008, pp. 35 – 44. Retrieved on March 10, 2011, from ftp://ftp.pitt.edu/users/d/r/druzdzel/iis08.pdf

[15] P. Myllymaki, "Advantages of Bayesian Networks in Data Mining and Knowledge Discovery", 2010. http://www.bayesit.com/docs/advantages.html

[16] G. Heinrich, "Parameter estimation for text analysis", Technical Note Version 2.4, vsonix GmbH and University of Leipzig, August, 2008. Retrieved on July 19, 2010 from http://www.arbylon.net/publications/text-est.pdf.

[17] R. G. Cowell, "Parameter Learning from Incomplete Data for Bayesian Networks", 1999. Retrieved from http://www.staff.city.ac.uk/~rgc/webpages/aistats99.pdf

[18] L. Oteniya, "Bayesian belief networks for dementia diagnosis and other applications: a comparison of hand-crafting and construction using a novel data driven technique", Unpublished PhD Thesis, Department of Computing Science, University of Stirling, Stirling, FK9 4LA, Scotland

[19] P. Spirtes, C. Glymour, and R. Scheines, "An algorithm for fast recovery of sparse causal graphs", *Social Science Computer Review*, vol. 9, 1991, pp. 62-72. Retrieved on July 22, 2010, from http://www.hss.cmu.edu/philosophy/techreports/15_Spirtes.pdf.

[20] F. Sahin, M. C. Yavuz, Z. Arnavut, and O. Uluyol, "Fault diagnosis for airplane engines using Bayesian networks and distributed particle swarm optimization", *Parallel Computing,* vol. 33, no. 2, 2007, pp. 124-143.
[21] D. M. Chickering, D. Geiger, and D. Heckerman, "Learning Bayesian Networks: Search Methods and Experimental Results", 1995. Retrieved from http://research.microsoft.com/en-us/um/people/dmax/publications/aistats95.pdf
[22] D. Heckerman, "A Tutorial on Learning With Bayesian Networks", Technical Report MSR-TR-95-06, USA: Microsoft Research. Retrieved from http://research.microsoft.com/pubs/69588/tr-95-06.pdf
[23] W. Lam and F. Bacchus, "Learning Bayesian belief networks: an approach based on the MDL principle", *Computational Intelligence,* vol. 10, 1994, pp. 269–293.
[24] H. Akaike, "A new look at the statistical model identification", *IEEE Transactions on Automatic Control*, vol. 19, no. 6, 1974, pp. 716–723.
[25] G. Schwarz, "Estimation the dimension of a model", *Annals of Statistics* vol. 6, 1978, pp. 462–464.
[26] D. Heckerman, D. Geiger, and D. M. Chickering, "Learning Bayesian Networks: The Combination of Knowledge and Statistical Data", vol. 20, 1995, pp. 197–243, Kluwer Academic Publishers, Hingham, MA, USA.
[27] W. Buntine, "Theory refinement on Bayesian networks", *Proceedings of the 7th Annual Conference on Uncertainty in Artificial Intelligence*, Los Angeles, CA, 1991, pp. 52 – 60.
[28] S. Yang and K. Chang, "Comparison of Score Metrics for Bayesian Network learning", *IEEE Transactions on Systems, Man, and Cybernetics*, Part A: Systems and Humans, vol. 32, no. 3, 2002, pp. 419-428. Retrieved on July 22, 2010 from http://volgenau.gmu.edu/~kchang/publications/journal_pdf/com_score_matrics.pdf.
[29] G. F. Cooper, "A Bayesian Method for Learning Belief Networks that Contain Hidden Variables", AAAI Technical Report WS-93-02, 1993. Retrieved from http://www.aaai.org/Papers/Workshops/1993/WS-93-02/WS93-02-011.pdf
[30] M. Forster, and E. Sober, "AIC Scores as Evidence – a Bayesian Interpretation", 2010. Retrieved from http://philosophy.wisc.edu/sober/forster%20and%20sober%20AIC%20Scores%20as%20Evidence%20jan%2028%202010.pdf
[31] D. M. Chickering, D. Geiger, and D. Heckerman, "Learning Bayesian networks is np-hard", Technical Report MSR-TR-94-17, Microsoft Research, November, 1994. Retrieved on July 22, 2010 from http://research.microsoft.com/apps/pubs/default.aspx?id=69598.
[32] R. W. Robinson, "Counting labelled acyclic digraphs", 1973. In F. Harary (Ed.), *New Directions in the Theory of Graphs*, New York: Academic Press, pp. 239-273.
[33] S. Anderson, D. Madigan, and M. Perlman, "A characterization of markov equivalence classes for acyclic digraphs", *Annals of Statistics*, vol. 25, 1997, pp. 505 -541.
[34] M. J. Druzdzel, "GeNIe: A development environment for graphical decision-analytic models", Proceedings of the Annual Symposium of the American Medical Informatics Association, Washington, D.C., 1999, pp. 1206. Retrieved from http://www.pitt.edu/~druzdzel/psfiles/amia99.pdf
[35] NSC (Norsys Software Corp), Netica-J Reference Manual (Version 3.25), 2008. http://www.norsys.com/netica-j/docs/NeticaJ_Man.pdf
[36] ARG (Automated Reasoning Group) (2004), SamIam. http://reasoning.cs.ucla.edu/samiam/
[37] D. M. Chickering, "The WinMine Toolkit", Technical Report: MSR-TR-2002-103, October 2002. Retrieved from http://research.microsoft.com/en-us/um/people/dmax/WinMine/WinMine.pdf
[38] M. Ramoni and P. Sebastiani, "Learning Bayesian networks from incomplete databases", KMi Technical Report KMi-TR-43, *Intelligent Data Analysis Journal*, vol. 2, no. 1, 1997.
[39] J. Pearl and S. Russell, "Bayesian Networks", November 2000. Retrieved February 12, 2010, from http://www.cs.berkeley.edu/~russell/papers/hbtnn-bn.ps
[40] MR (Microsoft Research), "WinMine Toolkit Tutorial:, Retrieved on February 02, 2011, from http://research.microsoft.com/en-us/um/people/dmax/WinMine/Tutorial/Tutorial.html.
[41] BNTT (Bayesian Network Toolbox Tutorial), "How to use the Bayes Net Toolbox", October, 2007. Retrieved on February 02, 2011 from http://bnt.googlecode.com/svn/trunk/docs/usage.html
[42] D. Pennock, "Evaluating probabilistic predictions", December, 2006. Retrieved from http://blog.oddhead.com/2006/12/26/evaluating-probabilistic-predictions/
[43] G. Blattenberger and F. Lad, "Separating the Brier Score into Calibration and Refinement Components: A Graphical Exposition", *The American Statistician*, vol. 39, no. 1, 1985, pp. 26-32. Retrieved from http://www.jstor.org/stable/pdfplus/2683902.pdf
[44] M. Morgan and M. Henrion, *Uncertainty: a guide to dealing with uncertainty in quantitative risk and policy analysis*, London: Cambridge University Press, 1990.
[45] P. Doshi, L. Greenwald, and J. Clarke, "Towards Effective Structure Learning for Large Bayesian Networks", AAAI Technical Report WS-02-14, 2002. Retrieved from http://www.aaai.org/Papers/Workshops/2002/WS-02-14/WS02-14-003.pdf
[46] NSC (Norsys Software Corp), "Advanced Topics: Testing nets with Cases", 2008. http://www.norsys.com/tutorials/netica/secD/tut_D2.htm
[47] I. J. Good, "Rational decisions", *Journal of the Royal Statistical Society,* vol. 14, 1952, pp. 107-114. In M. Roulston, "The Logarithmic Scoring Rule a.k.a. "ignorance"", Retrieved from http://www.cawcr.gov.au/bmrc/wefor/staff/eee/verif/Ignorance.html
[48] F. V. Jenson, *Bayesian Networks and Decision Graphs*. Springer, USA, 2001.
[49] M. S. Roulston and L. A. Smith, "Evaluating probabilistic forecasts using information theory", *Monthly Weather Review,* vol. 130, 2002, pp. 1653-1660.