RUHUNA JOURNAL OF SCIENCE
Vol. 1, September 2006, pp. 113–124
http://www.ruh.ac.lk/rjs/
ISSN 1800-279X

Inferencing design styles using Bayesian Networks

Aruna Lorensuhewa*, Binh Pham, and Shlomo Geva
Centre for information Technology Innovation, Faculty of Information Technology, Queensland University of Technology, Brisbane, Australia, s.lorensuhewa, b.pham, s.geva@qut.edu.au,

Reasoning with uncertain knowledge and belief has long been recognized as an important research issue in Artificial Intelligence (AI). Several methodologies have been proposed in the past, including knowledge-based systems, fuzzy sets, and probability theory. The probabilistic approach became popular mainly due to a knowledge representation framework called Bayesian networks. Bayesian networks have earned reputation of being powerful tools for modeling complex problem involving uncertain knowledge. Uncertain knowledge exists in domains such as medicine, law, geographical information systems and design as it is difficult to retrieve all knowledge and experience from experts. In design domain, experts believe that design style is an intangible concept and that its knowledge is difficult to be presented in a formal way. The aim of the research is to find ways to represent design style knowledge in Bayesian networks. We showed that these networks can be used for diagnosis (inferences) and classification of design style. The furniture design style is selected as an example domain, however the method can be used for any other domain.

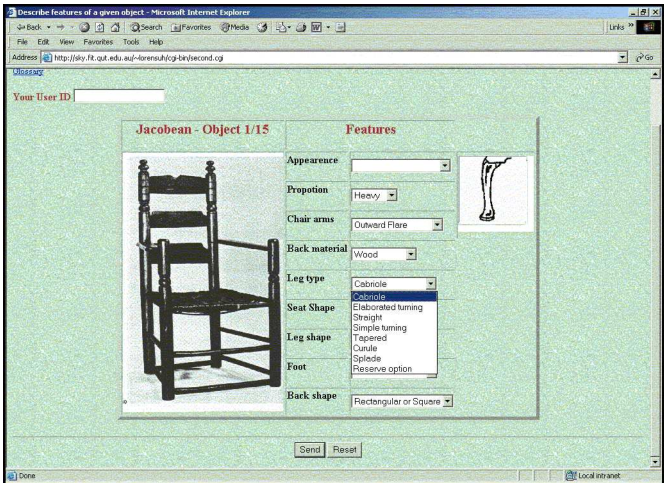
Key words: Bayesian networks, Classification, Design Style, SVM, C4.5, Data Mining

1. Introduction

Uncertain and fuzzy knowledge exists in domains where it is difficult to retrieve all knowledge and experience from experts. In the area of design, some experts believe that design style is an intangible concept and that its knowledge is difficult to be presented in a formal way. So far there has been no computer supported automatic techniques to assist novice designers in learning to distinguish design styles, to judge how similar a design is to a specific style or to learn suitable features for a required design by given a subset of features.

Machine learning and data mining techniques have been used to automatically extracted knowledge from unstructured information sources. Commonly-used algorithms include C4.5 (Quinlan 1993), Support Vector Machines (SVM) (Joachims 1999), Nearest Neighbor and Neural Networks. In our previous paper (Lorensuhewa 2003), we showed how these techniques can be used for classification task. Reasoning with uncertain knowledge and belief has long been recognized as an important research issue in AI. Several methodologies have been proposed in the past, including knowledge-based systems, fuzzy sets, and probability theory. The probabilistic

* Permanent Address: Department of Computer Science, University of Ruhuna, Matara, Sri Lanka.



Address http://sky.fit.qut.edu.au/~lorensuh/cgi-bin/second.cgi

Your User ID

Appearence

Propotion: Heavy

Chair arms: Outward Flare

Back material: Wood

Leg type: Cabriole

Seat Shape

Leg shape

Foot

Back shape: Rectangular or Square

Figure 1 Web based questionnaire of the experiment.

approach became popular mainly due to a knowledge representation framework called Bayesian Networks (BNs).

BNs provide a graphical representation of independencies amongst random variables (Korb 2003). A BN is a Directed Acyclic Graph (DAG) with nodes representing random variables and arc representing direct influence. The independence that is encoded in a BN means that each variable is independent of its non-descendents given its parents. To quantify the strengths of these associations, each node has a conditional-probability table that captures these associations among those random variables. BNs have been introduced in 1980s as formalism for representing and reasoning with models of problems involving uncertainty, adopting probability theory as a basic framework (Pearl 1988). Since the beginning of 1990s, researchers are exploring its capabilities for developing medical applications. The BN formalism offers a natural way to represent the uncertainties involved in medicine when dealing with diagnosis, treatment selection, planning, and prediction of prognosis. Similarly, we can use BN technique in design style domain.

The aim of this research is to develop a generic framework and methodologies that will enable the extraction of domain knowledge and information. We then represent this knowledge in BN and integrate with other expert knowledge and inference with the resulting model according to user requirements.

We collected an experimental data set for six different styles: Chippendale, Chippendale, Jacobean, Queen Anne, Sharon and William and Marry. A Web based questionnaire (Figure 1) is used to collect data from users and domain experts. In total, sixteen different features were examined and seven features were selected, including leg type, back shape, foot, etc. We also make use of the Connectedline Furniture Design Style Guide database (Connectedlines 1998) which is a commercially available database. This guide identifies and dates about 20 furniture styles and their distinctive features. This database is used as a source for expert knowledge.

The remainder of the paper is organized as follows: Section 2 provides the background for Bayesian network methodology. Section 3 describes how to construct Bayesian network from data for design style domain and how to use that for style classification. The method of integrating expert knowledge and how to increase the total classification accuracy is presented in Section 4. Section 5 discusses how to use Bayesian networks to inference about design style domain. The conclusion is given in the final section.

2. Bayesian Networks

Bayesian networks (or belief networks, casual networks) provide a graphical model for probabilistic relationships among a set of variables allows us to represent and reason about uncertain domains. The nodes in a BN represent a set of random variables from the domain. Random variables can be discrete or continuous. A set of directed arcs (or links) connects each pair of nodes, representing direct dependencies between variables (or in other word, casual relationships between variables). An arrow from node X to node Y means X is the parent of Y and X has a direct influence on Y. Each node has a conditional probability (for discrete variables) table to represent the strength of the relationships between variables. Each root node has a prior probability table. The only constraint on the arcs allowed in a BN is that there must not be any directed cycles.

2.1. Structure of the Bayesian Networks

Most commonly, BNs are considered to be the representations of joint probability distributions. There is a fundamental assumption that there is a useful underlying structure to the problem being modeled that can be captured with a BN, i.e., not every node is connected to every other node. If such a domain structure exists, a BN gives a more compact representation than simply describing the probability of every joint instantiation of all variables. The BN with few arcs or less parents for each node (sparse BNs) represents probability distributions in a computationally tractable way.

Consider a BN containing the n nodes, X_1 to X_n, taken in that order. A particular value in the joint distribution is represented by $P(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n)$, or more compactly, $P(x_1, x_2, \ldots, x_n)$. The chain rule of probability theory allows us to factorize joint probabilities so that

$$\begin{aligned} P(x_1, x_2, \ldots, x_n) &= P(x_1) \times P(x_2|x_1) \times \ldots \times P(x_n|x_1, \ldots, x_{n-1}) \\ &= \Pi_i P(x_n|x_1, \ldots, x_{n-1}) \end{aligned}$$

The structure of a BN implies that the value of a particular node is conditional only on the values of its parent nodes. This reduces to:

$$P(x_1, x_2, \ldots, x_n) = \Pi_i P(x_i|Parents(X_i))$$

provided that $Parents(X_i) \subseteq \{x_1, \ldots, x_{i-1}\}$

2.2. Building a Bayesian Network

There are a number of steps that a knowledge engineer must undertake when building a Bayesian network. These steps are: identifying the variables and its relevant values, finding the topological structure and specifying the conditional probability tables (Kord (2003)).

Identifying nodes and values: Firstly the variables of interest in the domain must be identified. In other words, the knowledge engineer needs to identify what the nodes represent and what values they take. In the case of discrete nodes, the following types can exist: Boolean nodes, ordered values or integer values.

Finding the structure: The structure of the network represents causes and effects relationships. It captures qualitative relationships between variables. In particular, two nodes should be connected directly if one affects or causes the other, with the arc indicating the direction of the effect. Usually expert knowledge is used to build the structure. For example, in the case of the medical diagnosis example, the knowledge engineer might ask the question "what factors affect a patient's chance of having cancer?" If the answer is "pollution and smoking", then arcs are added from the Pollution and Smoker node to the Cancer node. If the expert knowledge is not available, the structure should learn in another way. Learning the structure from data is a commonly used method for such a situation.

Specifying conditional probabilities: Once the structure of the BN is determined, the next step is to quantify the relationship between connected nodes. In the case of discrete variables, this is done by specifying conditional probability table (CPT) for each node.

2.3. Learning Structure and Parameters

BN were originally developed as knowledge representation formalism, with human experts their only resource. During the late 80's, people realized that the statistical foundations of Bayesian makes it possible to learn from data rather than from experts (Korb (2003)), and started to use both sources. Learning a BN means to learn the graphical model of dependencies (structure) and the conditional probability distributions (parameters).

There are two general approaches for constructing a graphical probabilistic model learning from data: the search and scoring methods and the dependency analysis methods (Cheng 1997). In the first approach, the algorithms view the learning problem as to search for a structure that can fit the data best. These algorithms start without any edges, and then use some search technique to add edges to the graph. After each change, they use some scoring scheme to compare the new and old structures. If the new structure is better than old one, they keep the newly added edge and another one. This process continues until the best structure is found. Several scoring methods have been used, namely Bayesian scoring method (Cooper (1992)) and Heckerman (1995)), entropy based method and minimum description length method (Bouckaert (1994)). Most of these algorithms need node ordering as the search space is very large.

In the second approach, the learning problem is viewed differently. The BN structure encodes a group of conditional independence relationships among the nodes, according to the concept of d-separation (Pearl (1988)). This suggests a way of learning the BN structure by identifying the conditional independence relationships among the nodes. These algorithms try to discover dependencies from the data, and then use these dependencies to infer the structure. The dependency relationships are measured by using a statistical test such as Chi-squared and mutual information tests. This type of algorithms are refereed to as CI-based algorithms or constraint-based algorithms (Cheng (1997)).

The parameters for a given structure can be learned by simply using the empirical conditional frequencies from the data by using corpus of complete data.

2.4. Inference in Bayesian networks

The basic task for any probabilistic inference system is to compute the posterior probability distribution for a set of query variables, given some observed events-that is, some assignment of values to a set of evidence variables. This task is called belief updating or probabilistic inference. Inference in Bayesian network is very flexible and useful for most of the application domains. In Bayesian inference, evidence can be entered about any node while beliefs in any other nodes are updated.

There are mainly two classes of existing inference algorithms, exact and approximate inference. Different algorithms are suited for different network structures and performance requirements. We have selected the most popular exact inference algorithm called junction tree (Jensen 1994). The junction tree algorithm is a process in two steps, transformation and propagation. The transformation builds an undirected junction tree from the Bayesian network. The second step, propagation, is where we propagate received evidence and make inference about variables using only the junction tree. Propagation is done by using message passing. This is implemented in Bayes Net Matlab Toolbox (Murphy (2002)) which we have used to conduct our experiments.

3. Construction of Bayesian Networks for Style Classification

This section describes how BNs are used as a knowledge representation tool to represent furniture design style knowledge. It also explains how BNs can be used for classification as well as for inference. The Bayes Net Matlab Toolbox (Murphy (2002)) was used for implementation. The furniture design style domain is selected as an example domain for this experiment and two different datasets were used for the experiment. The first dataset is collected from experts and users and the second dataset is artificially generated from Connectedline database.

3.1. Data Preparation

Data is selected for six different styles: Chippendale, Classical, Jacobean, Sharon, Queen Anne and William and Marry. A web-based questionnaire (Figure 1) was used to collect data from users and domain experts. In total, sixteen different features

Table 1 Listing of all the features and their encoded numbers

Feature	Encoded numbers for each feature type
Appearance	1,2,3,4,5
Proportion	1,2,3
Chair Arms	1,2,3,4,5,6,7,8
Back Material	1,2,3,4,5
Leg Type	1,2,3,4,5,6,7
Seat Shape	1,2,3,4
Foot	1,2,3,4,5,6,7,8,9,10,11,12,13,14
Back Shape	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20
Seat Material	1,2,3,4,5
Upholstery	1,2,3,4
Underbracing	1,2,3
Leg Shape	1,2
Leg Carved	1,2
Leg Curved	1,2
Leg Fluted	1,2
Style	1,2 ,3,4,5,6

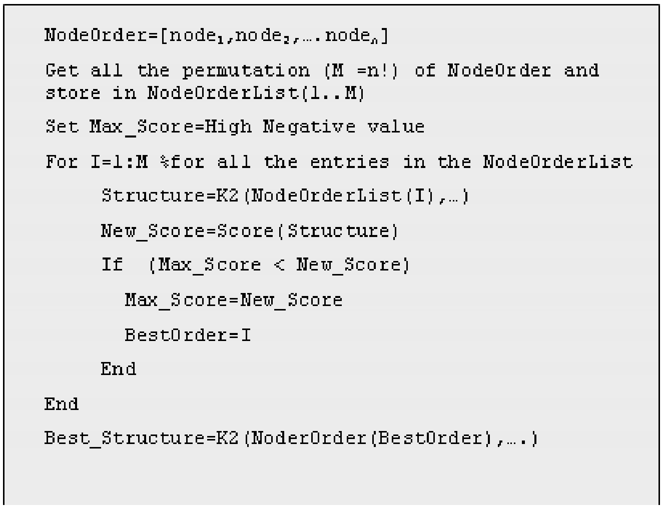
Table 2 The features of furniture design styles as described in Connecetdline database

Feature	Jacobean	William an Marry	Queen Anne	Chippendale	Hepplewhite	Sharon
Leg Type	4,3	1,2,3	1	1,3	3,5	3,4,5,6,7
Seat Shape	1	1	2,3	1,3,4	3	2,3
Foot Type	1,7,14	9,10,14	2,11,12,13,14	11,8,7.2	8,7,3	1,2,3
Back Shape	1,11	12	16,18,19	2,13,17	4,3	1,2,5,6,12
Leg Shape	1,2	1	1	1	1,2	1,2
Leg Fluted	1	1	1	1	2	2

were examined. These features are listed in Table 1, together with their encoded numbers. Each number in this table has a symbolic meaning. For example, Foot Type 1 is Lion Leg, Back Shape 16 is Fiddle Back, etc. Only subset of most significant features was selected for the experiment. In this paper, we will be labeling this data set as the 'observed' dataset.

The Connectedline furniture design style guide database (Connectedlines (1998)) is commercially available software for the Windows platform. This guide identifies and dates about 20 furniture styles and their distinctive features. This database is used to create an expert BN for this research. In the Connectedline database features of each style is described as in Table 2. There are many features described in the database, but we only selected the most important features which intersect with the features in the experimental dataset.

From this feature table, all the combinations of features in a specific style were artificially generated and stored in a database. Some design styles generated less amount of data because of a smaller number of corresponding combination of features. To generate an equal size of samples from each style, smaller samples were increased by duplicating each data instances in the samples several times. In this paper, we will call this data set the 'expert' data set.



```
NodeOrder=[$node_1$,$node_2$,….$node_n$]
Get all the permutation (M =n!) of NodeOrder and
store in NodeOrderList(1..M)
Set Max_Score=High Negative value
For I=1:M %for all the entries in the NodeOrderList
     Structure=K2(NodeOrderList(I),…)
     New_Score=Score(Structure)
     If  (Max_Score < New_Score)
       Max_Score=New_Score
       BestOrder=I
     End
End
Best_Structure=K2(NoderOrder(BestOrder),….)
```

Figure 2 An Algorithm to search for the best node order by using K2 algorithm and Bayesian score.

3.2. Learning Bayesian Network for Furniture Style Domain

We used the algorithm given by the Cooper and Herskovits (Cooper (1992)) to find the network structure from the observed dataset. Cooper and Herskovits showed that the best model, which maximizes the posterior probability of the network structure given the data, $P(B_s|D)$ also maximizes the joint probability, $P(B_s|D)$. They derived a function for this joint probability by using frequency of variable instantiation in the dataset. This function gives a direct way of measuring the goodness of fit between the model structure and the dataset and therefore it defines the fitness function (Bayesian score) for the model.

They use this fitness function to search the model space to find the good network structure. K2 requires fixed ordering of nodes, such that only the ‘upstream’ nodes of a node are considered as its candidate parents.

As we do not know the node order for the case of furniture design domain, we need to find out the correct node order to use the K2 to find the structure. We permutated all possible node order sequence and then sequentially search for the best order. We used the K2 algorithm to find a best structure for a given order sequence. The Bayesian score is used to compare two consecutive structures to find the best one. This simple technique of searching the best structure is given in Algorithm 1.

Different methods such as Genetic Algorithms can be used to find the best structure efficiently when the search space is bigger. After learning the structure, other parameters are also learned from the dataset.

Two Bayesian networks have learned from this method for the research experiments. The first BN has learned from the ‘observed’ data and it is labeled the ‘observed’ BN. The second BN has learned from ‘expert’ dataset and it is called the ‘expert’ BN.

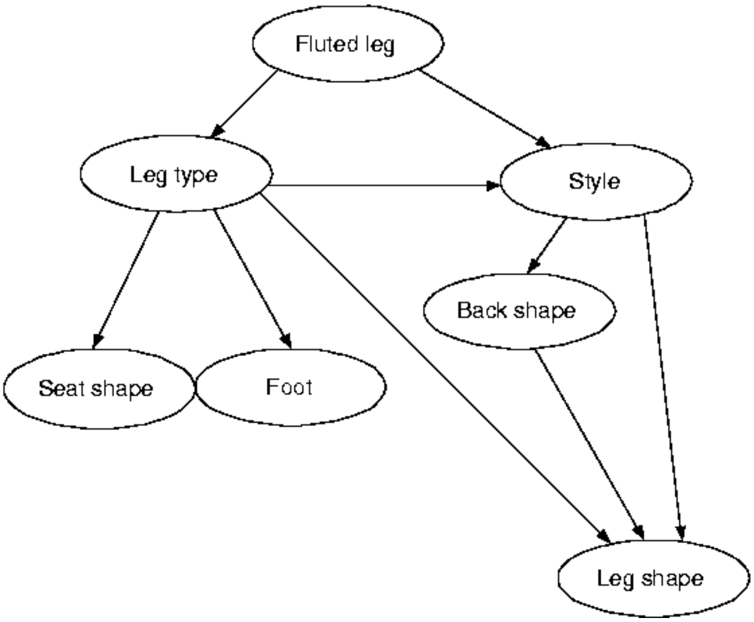


Figure 3 Bayesian Network Structure for fold 1.

3.3. Experimental Details and Summary of Results

Experiments were conducted with some subsets of features (seven features). The size of the 'observed' dataset is 120 instances. The 'observed' BN is constructed from 'observed' dataset and its accuracy is measured from the classification power of the BN. The target variable is the style node and the remaining six nodes are evidence nodes.

Cross validation is often used to estimate the generalization ability of a classifier where the amount of data is insufficient. Under cross validation, the available data is firstly divided into k disjointed sets. K models are then trained (in this case, k BNs and k CPDs), each one with different combination of (k-1) partitions and tested on the remaining partition. Throughout our experiments, we used 10-fold cross validation. Training dataset in each fold was used to find the best structure from Algorithm 1 and parameters (CPDs) for the selected structure. Instances in the test dataset which are relevant to the same fold are classified using the relevant BN. The Bayesian network structure relevant for fold one is given in Figure 2.

The 'expert' Bayesian network is constructed from the 'expert' dataset by a similar method. This BN is tested with the 'observed' dataset. The structure of the 'expert' Bayesian network is shown in Figure 3.

3.3.1. Comparison with the other techniques The experiments were conducted with three other techniques, SVM, Nearest Neighbour and C4.5 with the same 'observed' dataset using all the sixteen attributes. The original dataset with nominal attributes was tested with the Nearest Neighbour and C4.5 classifiers. The Nearest Neighbour classifier used exact matching as a distance metric. A windows-based software implementation of C4.5 (See 5) was used as a decision tree classifier.

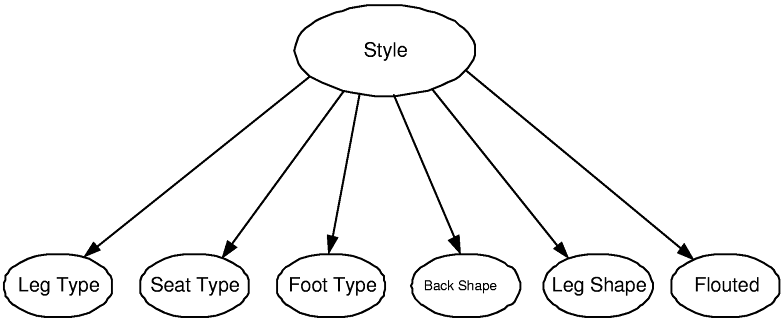


Figure 4 Bayesian network structure for 'expert' BN.

In both cases 10-fold cross validation is used for validation. The SVM classifier was used with the binary coded dataset. The original categorical (nominal) dataset with sixteen attributes is converted to a binary dataset with 100 binary attributes excluding the class attribute which has 6 different class numbers. In this case, six different binary classifiers were trained to classify six different styles separately. Each classifier determined if a given style attributes belonged to the corresponding style or not (binary classification). The classification decision for the entire ensemble of classifiers was based on the classifier giving the maximum output value (largest margin). In all three cases, ten-fold cross validation was used for validation. The summary of results is given in the Table 3.

Table 3 Comparison of classification accuracies of different techniques

Classifier	Data type	Classification Accuracy
Observe Bayesian Network	Categorical (7 attributes)	73.72±14.21
Expert Bayesian Network	Categorical (7 attributes)	68.31±10.32
Nearest Neighbour	Categorical (16 attributes)	85.59±9.49
SVM	Binary coded (100 attributes)	88.75±5.73
C4.5	Categorical (16 attributes)	76.50±3.70

4. Combining two Bayesian networks to improve overall classification accuracy

There are several methods of combining multiple classifiers (in this case two BN classifiers). Most commonly used technique is voting. Voting counts the number of classifiers that agree in their decision and accordingly decides the class to which the input pattern belongs. In this case, no accuracy or expertise is considered. According to our experimental results, the 'observed' BN is more accurate than expert BN. The 'observed' BN obtained 73% accuracy and the expert BN 68% accuracy. The Table 4 shows how both these networks behaved when they were tested separately using same test dataset.

From the Table 4, we can observe that, 36% (20.95±15.24) of the time both these network classifiers do not agree each other and only one is correct. It is difficult to

```
If (Prediction of Experimental BN == Prediction of
Expert Classifier)
    Get the prediction from either one
Else If (Maximum posterior probability of target
    variable for Expert BN -Maximum posterior
    probability of target variable for Observe
    BN)> Threshold Value (T)
    Get the prediction of Expert BN
Else
    Get the prediction of Observe BN
End
```

Figure 5 An algorithm to combine two Bayesian networks according to past experience.

Table 4 Behaviour of the two classifiers while classifing unseen test data

Percentage of cases both ‘observed’ BN and ‘expert’ BN are predicting correctly	52.38%
Percentage of cases bath ‘observed’ BN and ‘expert’ BN are not predicting correctly	11.43%
Percentage of cases only ‘observed’ BN is predicting correctly	20.95%
Percentage of cases only ‘expert’ BN is predicting correctly	15.24%

use the simple voting method to increase the overall accuracy by combining both Bayesian networks as the ‘observed’ BN is more accurate than the ‘expert’ BN. By analyzing the past results in Table 4, we construct following procedure (Algorithm 2) to combine the two classifiers to get a joint decision while classifying particular instances.

The task of finding a suitable Threshold value T for maximum classification accuracy is an important one. To determine a suitable Threshold value T, we conducted experiments with various values of T and measured the overall accuracy of the combined BN classifier using algorithm 2. T lies between 0$\sim$1. We plotted a graph of Threshold value (T) vs Classification accuracy of combined Bayesian network (Fig 4) and found that a threshold value range 0.35$\sim$0.4 for T gives the best accuracy of 83.64 ±8.35 for the combine network classifier.

5. Using BN to obtain Domain Knowledge

Given a complete BN model and defining both the structure and the conditional probabilities, we can begin to make prediction for any variable. If the values of some variables known (‘observed’), then the probabilities of the remaining variables (‘target’) can be calculated. In previous section we used BN for design style classification. In the style classification, we used the style variable as the target variable and all the other variables as observation variables. The same BN can be used further to learn about the furniture design style domain. For example, it can be used to learn the correct combination of parts for a specific design, or to learn possible suggestion for a missing (unknown) part of a particular design.

In the ‘observed’ or ‘expert’ Bayesian networks, we can define any node as a target node and all the other nodes as evidence (observe) nodes. For example, to complete a

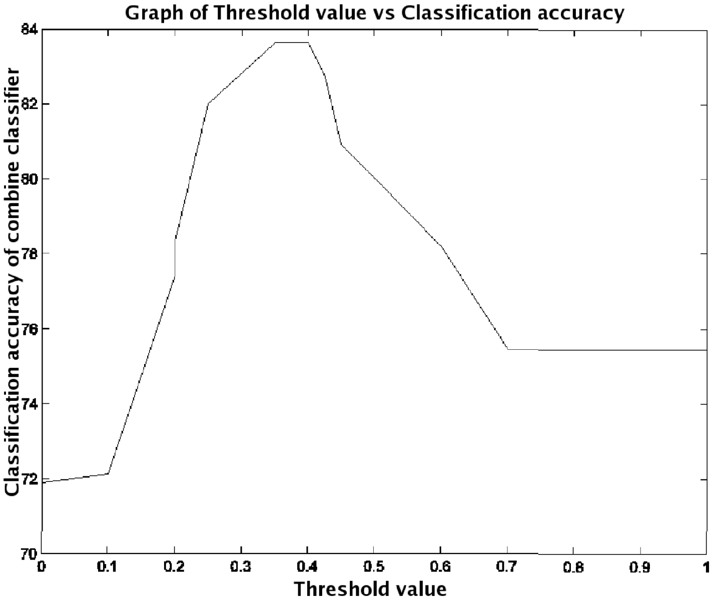


Figure 6 The graph of Threshold value Vs Classification accuracy of combined Bayesian Networks.

Table 5 Prediction accuarcy of target variables (Node) for the 'observed' Bayesian network

Observe Nodes	Target Node	Prediction Accuracy
Seat Shape, Foot, Back Shape, Leg Shape, Leg Type, Fluted, Style	Leg Type	81.36±14.36
Foot, Back Shape, Leg Shape, Leg Type, Leg Fluted, Style	Seat Shape	62.12±17.10
Seat Shape, Back Shape, Leg Shape, Leg Type, Leg Fluted, Style	Foot	53.18±13.22
Seat Shape, Foot, Leg Shape, Leg Type, Leg Fluted, Style	Back Shape	44.85±18.78
Seat Shape, Foot, Back Shape, Leg Type, Leg Fluted, Style	Leg Shape	90.91± 8.57
Seat Shape, Foot, Back Shape, Leg Shape, Leg Type, Style	Leg Fluted	96.36± 4.69
Seat Shape, Foot, Back Shape, Leg Shape, Leg Type, Leg Fluted	Style	73.72±14.21

particular design, a novice designer needs to find demonstrated deut the correct foot type. In this case, we select the foot type as the target node and all the other nodes as evidence nodes and update the posterior probability of the target node by giving evidence for all other evidence nodes. From the maximum posterior probability of foot node, we can suggest the most suitable foot type for that particular design. We have tested the prediction accuracy of each node in the 'observed' Bayesian network by giving evidence to all other nodes. The experimental results are shown in Table 3. Ten-fold cross validation was used throughout the experiment.

6. Conclusions

Bayesian networks provide a powerful tool and are able to represent and reason with uncertain knowledge. We have demonstrated that it can be used for diagnosis (inference) and classification of design style. We have found that the furniture design style can be classified with 73% of accuracy from the 'observed' Bayesian network.

The classification accuracy is slightly lower than the accuracies of SVM, C4.5 and Nearest Neighbour classifiers. The higher classification accuracy has been obtained by integrating the 'observed' BN with the 'expert' BN by using a simple algorithm based on the past performances of the two BNs.

Bayesian networks can be used not only for classification but also for inference on domain knowledge. This is the advantage of using Bayesian network over the other classifiers. We showed that higher accuracy can be achieved for most of the prediction nodes (Leg type, Leg fluted and Leg shape) other than the style node. The furniture design style is selected as an example domain, however the method can be used for other application domains such as medicine.

References

Bouckaert, R.R., 1994. Probabilistic Network Construction Using the Minimum Description Length Principle, Technical Report RUU-CS-94-27, Department of Computer Science, Utrecht University.

Cheng, J., D. Bell, and W. Liu., 1997. Learning Bayesian Networks from Data: An Efficient Approach Based on Information Theory. in Sixth ACM International Conference on Information and Knowledge Management. New York, USA.

Connectedlines, The On-Line Furniture Style Guide. 1998.
http://www.connectedlines.com/styleguide/index.htm.

Cooper, G.F. and E. Herskovits, A Bayesian Method for the Induction of Probabilistic Networks from Data. Machine Learning, 1992. 9: p. 309-347.

Heckerman, D., 1995. A Tutorial on Learning With Bayesian Networks. in Twelfth International Conference on Machine Learning. Tahoe City, California, USA: Morgan Kaufman.

Jensen, F., 1994. Implementation aspects of various propergation algorithms in Hugin. reserach Report R-94-2014, Department of Mathematics and Computer Science, Aalborg University: Denmark.

Joachims, T., 1999. SVM light Super Vector Machine.

Korb, K.B. and A.E. Nicholson, eds. Bayesian Artificial Intelligence. 2003, Chapman & Hall/ CRC Press UK.

Lorensuhewa, A., B. Pham, and S. Geva., 2003. Application of Machine Learning Techniques to Design Style Classification. in The 8th Australian and New Zealand Intelligent Information Systems Conference, Macquarie University, Sydney, Australia.

Murphy, K., 2002. Bayes Net Toolbox.
http://www.ai.mit.edu/ murphyk/Software/BNT/bnt.html.

Pearl, J., 1988. Probabilistic Reasoning in Intelligent Systems., San Mateo, California: Morgan Kaufman.

Quinlan, J.R., 1993. C4.5: Programs for Machine Learning: Morgan Kaufmann.